DEPARTAMENTO DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

TPE - Agricultura con drones

6 de mayo de 2016 — Algoritmos y Estructuras de Datos I

Grupo 15

Integrante	LU	Correo electrónico
Szperling, Sebastián	763/15	zebaszp@gmail.com
Barylko, Roni	750/15	ronibarylko@hotmail.com
Giudice, Carlos	694/15	Carlosr.giudice@gmail.com
López Segura, Florencia	759/13	fsegura@dc.uba.ar

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. Tipos

`tipo` Id = $\mathbb{Z}$;
`tipo` Carga = $\mathbb{Z}$;
`tipo` Ancho = $\mathbb{Z}$;
`tipo` Largo = $\mathbb{Z}$;
`tipo` Parcela = Cultivo, Granero, Casa ;
`tipo` Producto = Fertilizante, Plaguicida, PlaguicidaBajoConsumo, Herbicida, HerbicidaLargoAlcance ;
`tipo` EstadoCultivo = ReciénSembrado, EnCrecimiento, ListoParaCosechar, ConMaleza, ConPlaga, NoSensado ;

2. Campo

`tipo Campo` {
`observador dimensiones` (c: Campo) : (Ancho, Largo) ;
`observador contenido` (c: Campo, i, j: $\mathbb{Z}$) : Parcela ;
`requiere enRango` : $0 \leq i < prm(dimensiones(c)) \land 0 \leq j < sgd(dimensiones(c))$;

`invariante dimensionesValidas` : $prm(dimensiones(c)) > 0 \land sgd(dimensiones(c)) > 0$;
`invariante unaSolaCasa` : $|[(i,j)|i \leftarrow [0..prm(dimensiones(c))), j \leftarrow [0..sgd(dimensiones(c))), contenido(c,i,j) == Casa]| == 1$;
`invariante unSoloGranero` : $|[(i,j)|i \leftarrow [0..prm(dimensiones(c))), j \leftarrow [0..sgd(dimensiones(c))), contenido(c,i,j) == Granero]| == 1$;
`invariante algoDeCultivo` : $|[(i,j)|i \leftarrow [0..prm(dimensiones(c))), j \leftarrow [0..sgd(dimensiones(c))), contenido(c,i,j) == Cultivo]| \geq 1$;
`invariante posicionesAlcanzables` : $posicionesAlcanzablesEn100(c)$;
}

`aux posicionesAlcanzablesEn100` (c: Campo) : Bool = $alcanzableEn100(posicionGranero(c), prm(dimensiones(c)), sgd(dimensiones(c)))$;

`problema crearC` (posG, posC: ($\mathbb{Z}$, $\mathbb{Z}$)) = `res` : Campo {
`requiere puntosValidos` : $0 \leq \mathsf{prm}(posG) \land 0 \leq \mathsf{sgd}(posG) \land 0 \leq \mathsf{prm}(posC) \land 0 \leq \mathsf{sgd}(posC)$;
`requiere puntosDistintos` : $posG \neq posC$;
`requiere aMenosDe100` : $distancia(posG, posC) \leq 100$;
`requiere` $distancia((0,0), posG) \leq 100$;
`asegura` $enRango(dimensiones(res), prm(posG), sgd(posG)) \land (contenido(res, prm(posG), sgd(posG)) == Granero)$;
`asegura` $enRango(dimensiones(res), prm(posC), sgd(posC)) \land (contenido(res, prm(posC), sgd(posC)) == Casa)$;
}

`problema dimensionesC` (c: Campo) = `res` : (Ancho, Largo)) {
`asegura` $res == dimensiones(c)$;
}

`problema contenidoC` (c: Campo, i, j: $\mathbb{Z}$) = `res` : Parcela {
`requiere` $enRango(dimensiones(c), i, j)$;
`asegura` $res == contenido(c, i, j)$;
}

3. Drone

```
tipo Drone {
        observador id (d: Drone) : Id ;
        observador bateria (d: Drone) : Carga ;
        observador enVuelo (d: Drone) : Bool ;
        observador vueloRealizado (d: Drone) : [(ℤ, ℤ)] ;
        observador posicionActual (d: Drone) : (ℤ, ℤ) ;
        observador productosDisponibles (d: Drone) : [Producto] ;

        invariante vuelosOk :
            enVuelo(d) ⇒ (|vueloRealizado(d)| > 0 ∧ posicionActual(d) == vueloRealizado(d)_{|vueloRealizado(d)|−1} ∧
            posicionesPositivas(d) ∧ movimientosOK(d)) ∧ ¬enVuelo(d) ⇒ |vueloRealizado(d)| == 0 ;
        invariante bateriaOk : 0 ≤ bateria(d) ≤ 100 ;
}
aux posicionesPositivas (d: Drone) : Bool = (∀i ← [0.. |vueloRealizado(d)|))prm(vueloRealizado(d)_i) ≥ 0 ∧
sgd(vueloRealizado(d)_i ≥ 0 ;
aux movimientosOK (d: Drone) : Bool = (∀i ← [1.. |vueloRealizado(d)|))
prm(vueloRealizado(d)_i) == prm(vueloRealizado(d)_{i−1}) ∧ (sgd(vueloRealizado(d)_i) == sgd(vueloRealizado(d)_{i−1}) − 1 ∨
sgd(vueloRealizado(d)_i) == sgd(vueloRealizado(d)_{i−1}) + 1)
∨ sgd(vueloRealizado(d)_i) == sgd(vueloRealizado(d)_{i−1}) ∧ (prm(vueloRealizado(d)_i) == prm(vueloRealizado(d)_{i−1}) − 1
∨ prm(vueloRealizado(d)_{i−1}) == prm(vueloRealizado(d)_{i−1}) + 1) ;

problema crearD (i: ℤ, ps: [Producto]) = res : Drone {
        asegura bateria(res) == 100 ∧ id(res) == i ;
        asegura ¬enVuelo(res) ;
        asegura mismos(productosDisponibles(res), ps) ;
}

problema idD (d: Drone) = res : ℤ {
        asegura res == id(d) ;
}

problema bateriaD (d: Drone) = res : ℤ {
        asegura res == bateria(d) ;
}

problema enVueloD (d: Drone) = res : Bool {
        asegura res == enVuelo(d) ;
}

problema vueloRealizadoD (d: Drone) = res : [(ℤ, ℤ)] {
        requiere enVuelo(d) ;
        asegura mismos(res, [vueloRealizado(d)_i | i ← [1.. |vueloRealizado(d)|)] ;
}

problema posicionActualD (d: Drone) = res : (ℤ, ℤ) {
        asegura res == posicionActual(d) ;
}

problema productosDisponiblesD (d: Drone) = res : [Producto] {
        asegura mismos(res, productosDisponibles(d)) ;
}

problema vueloEscaleradoD (d: Drone) = res : Bool {
        asegura res == enVuelo(d) ∧ movimientoAlternado(d) ∧ movimientoUnidireccional(d) ;
}

problema vuelosCruzadosD (ds: [Drone]) = res : [((ℤ, ℤ), ℤ)] {
        requiere todosEnVuelo : (∀d ∈ ds)enVuelo(d) ;
        requiere vuelosDeIgualLargo : (∀i ∈ [1.. |ds|)) |vueloRealizado(ds_i)| == |vueloRealizado(ds_{i−1})| ;
        asegura (ds == [ ] ∧ res == [ ]) ∨ (mismos(res, [cruce | i ← [0.. |vueloRealizado(ds_0)|),
            cruce ← [(posicion, |[1|d ← ds, vueloRealizado(d)_i == posicion]|)|
            posicion ← posicionesSinRepetir([vueloRealizado(d)_i | d ← ds])], sgd(cruce) > 1])) ;
        asegura enOrden : (∀i ∈ [1.. |res|))sgd(res_i) ≥ sgd(res_{i−1}) ∨ (∀i ∈ [1.. |res|))sgd(res_i) ≤ sgd(res_{i−1}) ;
}
```

4. Sistema

tipo Sistema {
observador campo (s: Sistema) : Campo ;
observador estadoDelCultivo (s: Sistema, i, j: $\mathbb{Z}$) : EstadoCultivo ;
requiere $enRango(dimensiones(s), i, j) \wedge contenido(campo(s), i, j) == Cultivo$;
observador enjambreDrones (s: Sistema) : [Drone] ;

invariante identificadoresUnicos : $sinRepetidos([\, id(d) \mid d \leftarrow enjambreDrones(s) \,])$;
invariante unoPorParcela : $(\forall d, d' \leftarrow dronesEnVuelo(s), id(d) \neq id(d'))posicionActual(d) \neq posicionActual(d')$;
invariante siNoVuelanEstanEnGranero : $(\forall d \leftarrow enjambreDrones(s), \neg enVuelo(d))$
$posicionActual(d) == posicionGranero(campo(s))$;
invariante siEstanEnVueloElVueloEstaEnRango : $(\forall d \leftarrow dronesEnVuelo(s))(\forall v \leftarrow vueloRealizado(d))$
$enRango(dimensiones(campo(s), prm(v), sgd(v))$;
}

aux dronesEnVuelo (s: Sistema) : [Drone] $= [\, d \mid d \leftarrow enjambreDrones(s), enVuelo(d) \,]$;

problema crearS (c: Campo, ds: [Drone]) = res : Sistema {
requiere idsUnicos : $(\forall i, j \in listaIds(ds),\ i \neq j)\ listaIds(ds)_i \neq listaIds(ds)_j$;
requiere dronesEnGranero : $(\forall d \in ds)\ \neg enVuelo(d) \wedge\ posicionActual(d) == posicionGranero(c)$;
requiere bateriaLlena : $(\forall d \in ds)\ bateria(d) == 100$;
asegura cultivosNoSensados : $(\forall i \in [0...prm(dimensiones(c))),\ j \leftarrow [0...sgd(dimensiones(c))))$
$contenido(c, i, j) == Cultivo \longrightarrow estadoDelCultivo(res, i, j) == NoSensado$;
asegura $campo(res) == c$;
asegura mismosDrones : $|ds| == |enjambreDrones(res)| \wedge\ (\forall\ d1 \in ds)$
$(\exists\ d2 \in enjambreDrones(res))\ mismoDrone(d1, d2)$;
}

problema campoS (s: Sistema) = res : Campo {
asegura $res == campo(s)$;
}

problema estadoDelCultivoS (s: Sistema, i, j: $\mathbb{Z}$) = res : EstadoCultivo {
requiere $enRango(dimensiones(campo(s)), i, j)$;
requiere $contenido(campo(s), i, j) == Cultivo$;
asegura $res == estadoDelCultivo(s, i, j)$;
}

problema enjambreDronesS (s: Sistema) = res : [Drone] {
asegura $|enjambreDrones(s)| == |res| \wedge\ (\forall\ d1 \in enjambreDrones(s))\ (\exists\ d2 \in res)\ mismoDrone(d1, d2)$;
}

problema crecerS (s: Sistema) {
modifica s ;
asegura mismoCampo : $campo(\mathsf{pre}(s)) == campo(s)$;
asegura mismosDrones : $|enjambreDrones(\mathsf{pre}(s))| == |enjambreDrones(s)| \wedge\ (\forall\ d1 \in enjambreDrones(\mathsf{pre}(s)))$
$(\exists\ d2 \in enjambreDrones(s))\ mismoDrone(d1, d2)$;
asegura crecimientoCultivo : $(\forall i \in [0..\mathsf{prm}(dimensiones(campo(s))))(\forall j \in [0..\mathsf{sgd}(dimensiones(campo(s))))$
$contenido(campo(s), i, j) == Cultivo \longrightarrow$
$(estadoDelCultivo(\mathsf{pre}(s), i, j) == RecienSembrado \longrightarrow estadoDelCultivo(s, i, j) == EnCrecimiento$
$\wedge\ estadoDelCultivo(\mathsf{pre}(s), i, j) == EnCrecimiento \longrightarrow estadoDelCultivo(s, i, j) == ListoParaCosechar$
$\wedge\ (estadoDelCultivo(\mathsf{pre}(s), i, j) \neq RecienSembrado \wedge estadoDelCultivo(\mathsf{pre}(s), i, j) \neq EnCrecimiento)$
$\longrightarrow estadoDelCultivo(\mathsf{pre}(s), i, j) = estadoDelCultivo(s, i, j))$;
}

problema seVinoLaMalezaS (s: Sistema, ps: [($\mathbb{Z}$, $\mathbb{Z}$)]) {
requiere $(\forall (i, j) \in ps)\ enRango(dimensiones(campo(s)), i, j)$;
requiere $(\forall (i, j) \in ps)\ contenido(campo(s), i, j) == Cultivo$;
modifica s ;
asegura mismoCampo : $campo(\mathsf{pre}(s)) == campo(s)$;
asegura mismosDrones : $|enjambreDrones(\mathsf{pre}(s))| == |enjambreDrones(s)| \wedge\ (\forall\ d1 \in enjambreDrones(\mathsf{pre}(s)))$
$(\exists\ d2 \in enjambreDrones(s))\ mismoDrone(d1, d2)$;
asegura enmalezado : $(\forall i \in [0..\mathsf{prm}(dimensiones(campo(s))))(\forall j \in [0..\mathsf{sgd}(dimensiones(campo(s))))$
$\mathsf{en}((i, j), ps) \longrightarrow estadoDelCultivo(s, i, j) == ConMaleza$
$\wedge\ \neg(\mathsf{en}((i, j), ps)) \longrightarrow estadoDelCultivo(s, i, j) == estadoDelCultivo(pre(s), i, j)$;

}

problema seExpandePlagaS (s: Sistema) {
    modifica s;
    asegura mismoCampo : $campo(\mathsf{pre}(s)) == campo(s)$;
    asegura mismosDrones : $|enjambreDrones(\mathsf{pre}(s))| == |enjambreDrones(s)| \land (\forall\ d1 \in enjambreDrones(\mathsf{pre}(s)))$
        $(\exists\ d2 \in enjambreDrones(s))\ mismoDrone(d1, d2)$;
    asegura expansiónDePlaga : $(\forall i \in [0..\mathsf{prm}(dimensiones(campo(s))))(\forall j \in [0..\mathsf{sgd}(dimensiones(campo(s))))$
        $adyacenteAPlaga(s, i, j) \longrightarrow estadoDelCultivo(s, i, j) == ConPlaga$
        $\land \neg(adyacenteAPlaga(s, i, j)) \longrightarrow estadoDelCultivo(s, i, j) == estadoDelCultivo(pre(s), i, j)$;
}

problema despegarS (s: Sistema, d: Drone) {
    requiere $bateria(d) == 100$;
    requiere $\neg enVuelo(d)$;
    requiere $PosicionActual(d) == PosicionGranero(campo(s))$;
    requiere $(\exists e \in enjambreDrones(s))\ mismoDrone(d, e)$;
    modifica s;
    asegura mismoCampo : $campo(\mathsf{pre}(s)) == campo(s)$;
    asegura mismoEstadoCultivo : $(\forall i \in [0..\mathsf{prm}(dimensiones(campo(s)))))(\forall j \in [0..\mathsf{sgd}(dimensiones(campo(s))))$
        $contenido(campo(s), i, j) \neq Cultivo \lor estadoDelCultivo(s, i, j) == estadoDelCultivo(\mathsf{pre}(s), i, j))$;
    asegura mismosOtrosDrones : $|enjambreDrones(\mathsf{pre}(s))| == |enjambreDrones(s)| \land$
        $(\forall d1 \in enjambreDrones(\mathsf{pre}(s)),\ id(d1) \neq id(d))\ (\exists d2 \in enjambreDrones(s))\ mismoDrone(d1, d2)$;
    asegura elDroneDespega : $(\exists e \in enjambreDrones(s))\ id(e) == id(d)\ \land posicionActual(e) == posicionActual(d)$
        $\land\ bateria(e) == bateria(d)\ \land\ mismos(productosDisponibles(e), productosDisponibles(d))$
        $\land\ enVuelo(e)\ \land\ vueloRealizado(e) == [posicionActual]$;
}

problema listoParaCosecharS (s: Sistema) = res : Bool {
    asegura $res == (|[1|(i, j) \leftarrow cultivos(s), estadoDelCultivo(s, i, j) == ListoParaCosechar]| \div |cultivos(s)| \geq 0{,}9)$;
}

problema aterrizarYCargarBateriaS (s: Sistema, b: $\mathbb{Z}$) {
    requiere $0 < b \leq 100$;
    modifica s;
    asegura mismoCampo : $campo(\mathsf{pre}(s)) == campo(s)$;
    asegura mismoEstadoCultivo : $(\forall i \in [0..\mathsf{prm}(dimensiones(campo(s)))))(\forall j \in [0..\mathsf{sgd}(dimensiones(campo(s))))$
        $contenido(campo(s), i, j) \neq Cultivo \lor estadoDelCultivo(s, i, j) == estadoDelCultivo(pre(s), i, j))$;
    asegura mismosDrones : $|enjambreDrones(s)| == |enjambreDrones(\mathsf{pre}(s))|$;
    asegura cambiosADrones : $(\forall d \in enjambreDrones(pre(s)))(\exists e \in enjambreDrones(s))$
        $(bateria(d) \leq b \longrightarrow (id(e) == id(d)\ \land\ bateria(e) == 100\ \land\ \neg enVuelo(e)$
        $\land\ posicionActual(e) == posicionGranero(campo(s))\ \land\ vueloRealizado(d) == [\,]$
        $\land\ mismos(productosDisponibles(e), productosDisponibles(d))))$
        $\land (bateria(d) > b \longrightarrow mismoDrone(d, e))$;
}

problema fertilizarPorFilas (s: Sistema) {
    requiere maxUnDronPorFila : $(\forall j \in [0..\mathsf{sgd}(dimension(campo(s))))$
        $|[1|i \leftarrow [0..\mathsf{prm}(dimension(campo(s)))), \mathsf{en}((i, j), posicionesDronesEnVuelo(s))]| < 2$;
    modifica s;
    asegura mismoCampo : $campo(\mathsf{pre}(s)) == campo(s)$;
    asegura mismosDrones : $|enjambreDrones(s)| == |enjambreDrones(\mathsf{pre}(s))|$
        $\land (\forall d \in enjambreDrones(\mathsf{pre}(s)))(\exists e \in enjambreDrones(s))id(e) == id(d)$;
    asegura soloCambiarSiEnVuelo : $(\forall d \in enjambreDrones(\mathsf{pre}(s)))(\exists e \in enjambreDrones(s), id(e) == id(d))$
        $enVuelo(d) == enVuelo(e) \land \neg enVuelo(d) \longrightarrow mismoDrone(d, e)$;
    asegura vuelosAlOeste : $(\forall d \in enjambreDrones(\mathsf{pre}(s)), enVuelo(d))(\exists e \in enjambreDrones(s), id(e) == id(d))$
        $vueloRealizado(e) == vueloRealizado(d) + +[$
        $(\mathsf{prm}(vueloRealizado(d)_{|vueloRealizado(d)|-1}) - i, \mathsf{sgd}(vueloRealizado(d)_{|vueloRealizado(d)|-1}))$
        $|\ i \rightarrow [1..\,|vueloRealizado(e)| - |vueloRealizado(d)|]]$
        $\land\ posicionActual(e) == vueloRealizado(e)_{|vueloRealizado(e)|-1} \land \mathsf{prm}(posicionActual(e)) \geq 0$;
    asegura gastoDeBateria : $(\forall d \in enjambreDrones(\mathsf{pre}(s)))(\exists e \in enjambreDrones(s), id(e) == id(d))$
        $bateria(e) \geq 0 \land bateria(e) == (bateria(d) + |vueloDiferente(e, d)|)$;
    asegura unFertilizantePorCultivo : $(\forall d \in enjambreDrones(\mathsf{pre}(s)))(\exists e \in enjambreDrones(s), id(e) == id(d))$
        $mismos(productosDisponibles(d), productosDisponibles(e) + +[Fertilizante\ |\ i \leftarrow [0..\,|vueloDiferente(e, d)|)])$;

asegura soloCultivos : $(\forall d \in enjambreDrones(\mathsf{pre}(s)))(\exists e \in enjambreDrones(s), id(e) == id(d))$
$(\forall (i,j) \in vueloDiferente(e,d))contenido(campo(s),i,j) == Cultivo$;
asegura soloViajaPorCultivos : $(\forall d \in enjambreDrones(\mathsf{pre}(s)))$
$(\exists e \in enjambreDrones(s), id(e) == id(d))(\forall i \in [0..|vueloDiferente(e,d)|-1))$
$contenido(campo(s), \mathsf{prm}(vueloDiferente(e,d)_i), \mathsf{sgd}(vueloDiferente(e,d)_i)) == Cultivo$;
asegura condicionDeFinalizacion : $((\forall d \in enjambreDrones(s), enVuelo(d))$
$bateria(d) == 0 \vee \neg(\mathsf{en}(Fertilizante, productosDisponibles(d)))$
$\vee \mathsf{prm}(vueloRealizado(d)_{|vueloRealizado(d)|-1}) == 0)$
$\vee contenido(campo(s), \mathsf{prm}(posicionActual(d)), \mathsf{sgd}(posicionActual(d))) \neq Cultivo$;
asegura posicionesNoFertilizadas : $(\forall\ pos \in [(x,y) \mid x \to [0..\mathsf{prm}(dimensiones(campo(s))))$
$y \to [0..\mathsf{sgd}(dimensiones(campo(s)))), contenido(c,x,y) == Cultivo$
$\wedge \neg en((x,y), posicionesFertilizadas(s, pre(s)))])$
$estadoDeCultivo(s,x,y) == estadoDeCultivo(\mathsf{pre}(s),x,y)$;
asegura fertilizarCampo : $(\forall d \in enjambreDrones(\mathsf{pre}(s)))$
$(\exists\ e \in enjambreDrones(s), id(e) == id(d))(\forall (i,j) \in vueloDiferente(e,d))$
$((\mathsf{en}(estadoDelCultivo(\mathsf{pre}(s),i,j), [RecienSembrado, EnCrecimiento])$
$\longrightarrow estadoDelCultivo(s,i,j) == ListoParaCosechar)$
$\wedge (\neg\mathsf{en}(estadoDelCultivo(\mathsf{pre}(s),i,j), [RecienSembrado, EnCrecimiento])$
$\longrightarrow estadoDelCultivo(s,i,j) == estadoDelCultivo(\mathsf{pre}(s),i,j))$;
}

problema volarYSensarS (s: Sistema, d: Drone) {
requiere puedeMoverse : $bateria(d) > 0$;
requiere $enVuelo(d)$;
requiere $(\exists e \in enjambreDrones(s))mismoDrone(d,e)$;
modifica s ;
asegura mismoCampo : $campo(\mathsf{pre}(s)) == campo(s)$;
asegura mismosDrones : $|enjambreDrones(\mathsf{pre}(s))| == |enjambreDrones(s)| \wedge$
$(\forall d1 \in enjambreDrones(\mathsf{pre}(s)), \neg mismoDrone(d,d1))(\exists d2 \in enjambreDrones(s))mismoDrone(d1,d2)$;
asegura dronEnEnjambre : $(\exists e \in enjambreDrones(s))id(d) == id(e) \wedge enVuelo(e)$;
asegura aParcelaAdyacente : $(\exists e \in enjambreDrones(s), id(d) == id(e))$
$|vueloDiferente(e,d)| == 1 \wedge posicionActual(e) == vueloDiferente(e,d)_0$
$\wedge distancia(posicionActual(d), posicionActual(e)) == 1$
$\wedge enRango(dimensiones(campo(s)), \mathsf{prm}(posicionActual(e)), \mathsf{sgd}(posicionActual(e)))$;
asegura sensado : $(\exists e \in enjambreDrones(s), id(d) == id(e))$
$estadoDelCultivo(\mathsf{pre}(s), \mathsf{prm}(posicionActual(e)), \mathsf{sgd}(posicionActual(e))) == NoSensado$
$\longrightarrow (estadoDelCultivo(s, \mathsf{prm}(posicionActual(e)), \mathsf{sgd}(posicionActual(e))) \neq NoSensado \wedge noHaceNada(e,d)$;
asegura siEstaBienNoHaceNada : $(\exists e \in enjambreDrones(s), id(d) == id(e))$
$\neg\mathsf{en}(estadoDelCultivo(\mathsf{pre}(s), \mathsf{prm}(posicionActual(e)), \mathsf{sgd}(posicionActual(e))),$
$[ConMaleza, ConPlaga, NoSensado])$
$\longrightarrow (noHaceNada(e,d) \wedge estadoDelCultivo(\mathsf{pre}(s), \mathsf{prm}(posicionActual(e)), \mathsf{sgd}(posicionActual(e)))$
$== estadoDelCultivo(s, \mathsf{prm}(posicionActual(e)), \mathsf{sgd}(posicionActual(e))))$;
asegura mataMalezas : $(\exists e \in enjambreDrones(s), id(d) == id(e))$
$estadoDelCultivo(\mathsf{pre}(s), \mathsf{prm}(posicionActual(e)), \mathsf{sgd}(posicionActual(e))) == ConMaleza$
$\longrightarrow (usaAlgunHerbicida(e,d)$
$\wedge estadoDelCultivo(s, \mathsf{prm}(posicionActual(e)), \mathsf{sgd}(posicionActual(e))) == RecienSembrado)$
$\vee ((\neg tieneHerbicida(d) \vee bateria(d) < 6)$
$\wedge noHaceNada(e,d) \wedge estadoDelCultivo(s, \mathsf{prm}(posicionActual(e)), \mathsf{sgd}(posicionActual(e))) == ConMaleza))$;
asegura mataPlagas : $(\exists e \in enjambreDrones(s), id(d) == id(e))$
$estadoDelCultivo(\mathsf{pre}(s), \mathsf{prm}(posicionActual(e)), \mathsf{sgd}(posicionActual(e))) == ConPlaga$
$\longrightarrow (usaPlaguicidaComun(e,d) \wedge estadoDelCultivo(s, \mathsf{prm}(posicionActual(e)), \mathsf{sgd}(posicionActual(e)))$
$== RecienSembrado) \vee (usaPlaguicidaBajoConsumo(e,d)$
$\wedge estadoDelCultivo(s, \mathsf{prm}(posicionActual(e)), \mathsf{sgd}(posicionActual(e))) == RecienSembrado)$
$\vee ((tienePlaguicida(d) \wedge ((\neg tienePlaguicidaBajoConsumo(d) \wedge bateria(d) < 11)$
$\vee (tienePlaguicidaBajoConsumo(d) \wedge bateria(d) < 6)))$
$\vee (\neg tienePlaguicida(d) \wedge \neg tienePlaguicidaBajoConsumo(d))) \wedge noHaceNada(e,d)$
$\wedge estadoDelCultivo(s, \mathsf{prm}(posicionActual(e)), \mathsf{sgd}(posicionActual(e))) == ConPlaga))$;
asegura herbicidaLargoAlcance : $(\exists e \in enjambreDrones(s), id(d) == id(e))$
$mismos(productosDisponibles(d), HerbicidaLargoAlcance : productosDisponibles(e))$
$\longrightarrow ((\forall (x,y) \in cultivosAdyacentes(posicionActual(e), s))(estadoDelCultivo(\mathsf{pre}(s),x,y) == ConMaleza$
$\wedge estadoDelCultivo(s,x,y) == RecienSembrado) \vee (estadoDelCultivo(\mathsf{pre}(s),x,y) \neq ConMaleza$
$\wedge estadoDelCultivo(s,x,y) == estadoDelCultivo(\mathsf{pre}(s),x,y)))$;

asegura posicionesNoTocadas : $(\exists e \in enjambreDrones(s), id(e) == id(d))(\forall i \in [0..\mathsf{prm}(dimensiones(campo(s))))$
$(\forall j \in [0..\mathsf{sgd}(dimensiones(campo(s))), posicionActual(e) \neg (i,j))$
$(mismos(productosDisponibles(d), HerbicidaLargoAlcance : productosDisponibles(e))$
$\wedge\, distancia((i,j), posicionActual(e)) == 1) \vee (estadoDelCultivo(s,i,j) == estadoDelCultivo(\mathsf{pre}(s),i,j))$;
}

5. Ejercicios Adicionales

problema recargarProductos (s: Sistema, id: Id, ps: [Producto]) {
  requiere $(\exists d \in enjambreDrones(s))\ id(d) == id\ \wedge\ \neg enVuelo(d)\ \wedge\ bateria(d) == 100$;
  modifica s;
  asegura mismoCampo : $campo(\mathsf{pre}(s)) == campo(s)$;
  asegura mismoEstadoCultivo : $(\forall i \in [0..\mathsf{prm}(dimensiones(campo(s)))])(\forall j \in [0..\mathsf{sgd}(dimensiones(campo(s)))])$
    $contenido(campo(s), i, j) \neq Cultivo \vee estadoDelCultivo(s, i, j) == estadoDelCultivo(\mathsf{pre}(s), i, j))$;
  asegura mismosDrones : $|enjambreDrones(s)| == |enjambreDrones(\mathsf{pre}(s))| \wedge$
    $(\forall d \in enjambreDrones(\mathsf{pre}(s)), id(d) \neq id)\ (\exists e \in enjambreDrones(s))\ mismoDrone(d, e)$;
  asegura dronDeRecarga : $(\exists d \in enjambreDrones(s))(\exists e \in enjambreDrones(\mathsf{pre}(s)), id(d) == id(e))$
    $id(d) == id \wedge bateria(d) == 100 \wedge \neg enVuelo(d) \wedge vueloRealizado(d) == [\,]$
    $\wedge\ posicionActual(d) == posicionGranero(campo(s))$;
  asegura $(\exists d \in enjambreDrones(s))(\exists e \in enjambreDrones(\mathsf{pre}(s)), id(d) == id(e))$
    $losNuevosProductosDeLaLista(d, e, ps)\ \wedge\ 100 \geq costoEnergiaProductos(d)$
    $\wedge \neg en(Fertilizante, productosNoRecargados(d, e, ps))\ \wedge\ (costoEnergiaProductos(d) \leq 95)$
    $\longrightarrow (\neg en(Herbicida, productosNoRecargados(d, e, ps))$
    $\wedge \neg en(HerbicidaLargoAlcance, productosNoRecargados(d, e, ps))$
    $\wedge \neg en(PlaguicidaBajoConsumo, productosNoRecargados(d, e, ps)))$
    $\wedge\ (costoEnergiaProductos(d) \leq 90) \longrightarrow (\neg en(Plaguicida, productosNoRecargados(d, e, ps)))$;
}

problema vuelanEnFormacion (s: Sistema) = res : Bool {
  requiere $(\forall d \in enjambreDrones(s))\ |vueloRealizado(d)| > 3$;
  requiere $|enjambreDrones(s)| \geq 2$;
  asegura $res == (|enjambreDrones(s)| < |dronesEnFormacion(s)| \times 2)$;
}

problema droneAward (s: Sistema) = res : Drone {
  requiere $|enjambreDrones(s)| \geq 1$;
  asegura $mismoDrone(res, menorId(menosProductos(masParcelas(s))))$;
}

6. Funciones Auxiliares

6.1. Campo

`aux posicionGranero` (c: Campo) : ($\mathbb{Z}$, $\mathbb{Z}$) = $[(i,j) \mid i \leftarrow [0..prm(dimensiones(c))), j \leftarrow [0..sgd(dimensiones(c))), contenido(c,i,j) == Granero]_0$;

`aux posicionCasa` (c: Campo) : ($\mathbb{Z}$, $\mathbb{Z}$) = $[(i,j) \mid i \leftarrow [0..prm(dimensiones(c))), j \leftarrow [0..sgd(dimensiones(c))), contenido(c,i,j) == Casa]_0$;

6.2. Drone

`aux movimientoAlternado` (d: Drone) : Bool = $(\forall i \in [2..|vueloRealizado(d)|))$
$\mathsf{prm}(vueloRealizado(d)_i) == \mathsf{prm}(vueloRealizado(d)_{i-1}) \Leftrightarrow |\mathsf{prm}(vueloRealizado(d)_i) - \mathsf{prm}(vueloRealizado(d)_{i-2})| == 1 \wedge$
$\mathsf{sgd}(vueloRealizado(d)_i) == \mathsf{sgd}(vueloRealizado(d)_{i-1}) \Leftrightarrow |\mathsf{sgd}(vueloRealizado(d)_i) - \mathsf{sgd}(vueloRealizado(d)_{i-2})| == 1$;

`aux movimientoUnidireccional` (d: Drone) : Bool = $(\forall d \in [3..|vueloRealizado(d))|)$
$|\mathsf{prm}(vueloRealizado(d)_i) - \mathsf{prm}(vueloRealizado(d)_{i-1})| == 1 \iff$
$|\mathsf{prm}(vueloRealizado(d)_i) - \mathsf{prm}(vueloRealizado(d)_{i-3})| == 2 \wedge$
$|\mathsf{sgd}(vueloRealizado(d)_i) - \mathsf{sgd}(vueloRealizado(d)_{i-1})| == 1 \iff$
$|\mathsf{sgd}(vueloRealizado(d)_i) - \mathsf{sgd}(vueloRealizado(d)_{i-3})| == 2$;

`aux posicionesSinRepetir` (ps: [($\mathbb{Z}$, $\mathbb{Z}$)]) : [($\mathbb{Z}$, $\mathbb{Z}$)] = $[ps_i \mid i \leftarrow [0..|ps|), (\forall j \in [0..i))\ ps_i \neq ps_j]$;

6.3. Sistema

`aux mismoDrone` (d,e: Drone) : Bool = $id(d) == id(e) \wedge\ bateria(d) == bateria(e)$
$\wedge\ posicionActual(d) == posicionActual(e) \wedge\ mismos(productosDisponibles(d), productosDisponibles(e))$
$\wedge\ vueloRealizado(d) == vueloRealizado(e) \wedge\ enVuelo(d) == enVuelo(e)$;

`aux adyacenteAPlaga` (s: Sistema, (i,j): $\mathbb{Z}$) : Bool =
$\mathsf{en}(ConPlaga, [estadoDelCultivo(s,i,j) | (i,j) \leftarrow cultivosAdyacentes((i,j),s)])$;

`aux cultivosAdyacentes` ((i,j):($\mathbb{Z}$,$\mathbb{Z}$), s: Sistema) : [($\mathbb{Z}$, $\mathbb{Z}$)] = $[(x,y) | (x,y) \leftarrow [(i,j+1),(i+1,j),(i,j-1),(i-1,j)], enRango(dimensiones(campo(s)),x,y) \wedge contenido(campo(s),x,y) == Cultivo]$;

`aux cultivos` (s: Sistema) : [($\mathbb{Z}$, $\mathbb{Z}$)] = $[(i,j) | i \leftarrow [0..\mathsf{prm}(dimensiones(campo(s))))$
$j \leftarrow [0..\mathsf{sgd}(dimensiones(campo(s)))), contenido(campo(s),i,j) == Cultivo]$;

`aux posicionesDronesEnVuelo` (s: Sistema) : [($\mathbb{Z}$, $\mathbb{Z}$)] = $[posicionActual(d) | d \leftarrow enjambreDrones(s), enVuelo(d)]$;

`aux vueloDiferente` (e,d: Drone) : [($\mathbb{Z}$, $\mathbb{Z}$)] = $[vueloRealizado(e)_i | i \leftarrow [|vueloRealizado(d)| .. |vueloRealizado(e)|)]$;

`aux noHaceNada` (e,d: Drone) : Bool = $mismos(productosDisponibles(d), productosDisponibles(e))$
$\wedge\ bateria(d) - 1 == bateria(e)$;

`aux usaAlgunHerbicida` (e,d: Drone) : Bool = $(mismos(productosDisponibles(d),$
$Herbicida : productosDisponibles(e)) \vee mismos(productosDisponibles(d),$
$HerbicidaLargoAlcance : productosDisponibles(e))) \wedge bateria(e) == bateria(d) - 6$;

`aux tieneHerbicida` (d: Drone) : Bool = $\mathsf{en}(Herbicida, productosDisponibles(d))$
$\vee\ \mathsf{en}(HerbicidaLargoAlcance, productosDisponibles(d))$;

`aux usaPlaguicidaComun` (e,d: Drone) : Bool = $mismos(productosDisponibles(d),$
$Plaguicida : productosDisponibles(e)) \wedge bateria(e) == bateria(d) - 11$;

`aux usaPlaguicidaBajoConsumo` (e,d: Drone) : Bool = $mismos(productosDisponibles(d), PlaguicidaBajoConsumo :$
$productosDisponibles(e)) \wedge bateria(e) == bateria(d) - 6$;

`aux tienePlaguicida` (d: Drone) : Bool = $\mathsf{en}(Plaguicida, productosDisponibles(d))$;

`aux posicionesFertilizadas` (s,pres: Sistema) : [($\mathbb{Z}$,$\mathbb{Z}$)] = $[pos | d \leftarrow pres, e \leftarrow s, pos \leftarrow vueloDiferente(e,d), id(d) == id(e) \wedge enVuelo(e)]$;

6.4. Ejercicios Adicionales

`aux listaIds` (ds: [Drone]) : [$\mathbb{Z}$] = $[id(d) \mid d \leftarrow ds]$;

`aux losNuevosProductosDeLaLista` (d,e: Drone, ps: [Producto]) : Bool = $(\forall p1 \in productosDisponibles(d))$
$cantProducto(productosDisponibles(d), p1) - cantProducto(productosDisponibles(e), p1) \leq cantProducto(ps, p1)$;

`aux productosNoRecargados` (d,e: Drone, ps: [Producto]) : [Producto] =
$[Fertilizante | i \leftarrow [1, cantProdNoRecargados(d,e,ps,Fertilizante)]]$
$++[Plaguicida | i \leftarrow [1, cantProdNoRecargados(d,e,ps,Plaguicida)]]$
$++[PlaguicidaBajoConsumo | i \leftarrow [1, cantProdNoRecargados(d,e,ps,PlaguicidaBajoConsumo)]]$
$++[Herbicida | i \leftarrow [1, cantProdNoRecargados(d,e,ps,Herbicida)]]$
$++[HerbicidaLargoAlcance | i \leftarrow [1, cantProdNoRecargados(d,e,ps,HerbicidaLargoAlcance)]]$;

`aux cantProdNoRecargados` (d,e: Drone, ps: [Producto], p: Producto) : $\mathbb{Z}$ =
$cantProducto(ps,p) - cantProducto(productosDisponibles(d),p) + cantProducto(productosDisponibles(e),p)$;

`aux cantProducto` (ps: [Producto], p1: Producto) : $\mathbb{Z}$ = $|[1 \mid p \leftarrow ps, p == p1]|$;

`aux costoEnergiaProductos` (d:Drone) : $\mathbb{Z} = \sum[10 \mid p \leftarrow productosDisponibles(d), p == Plaguicida] + \sum[5 \mid p \leftarrow productosDisponibles(d), p == PlaguicidaBajoConsumo \vee p == Herbicida \vee p == HerbicidaLargoAlcance]$;

`aux dronesEnFormacion` (s:Sistema) : [Drone] $= [d \mid d \leftarrow enjambreDrones(s), enVuelo(d) \wedge (\forall i \in [|vueloRealizado(d)| - 3.. |vueloRealizado(d)|)(\exists e \in enjambreDrones(s))\ estanVolandoAlLado(d, e, i)]$;

`aux estanVolandoAlLado` (d,e:Drone, i:$\mathbb{Z}$) : Bool $= distancia(vueloRealizado(d)_i, vueloRealizado(e)_i) == 1$;

`aux menorId` (ds:[Drone]) : Drone $= cab[d1 \mid d1 \leftarrow ds, (\forall d2 \in ds) id(d1) \leq id(d2)]$;

`aux menosProductos` (ds:[Drone]) : [Drone] $= [d1 \mid d1 \leftarrow ds, (\forall d2 \in ds)\, |productosDisponibles(d1)| \leq |productosDisponibles(d2)|]$;

`aux masParcelas` (s:Sistema) : [Drone] $= [d1 \mid d1 \leftarrow enjambreDrones(s), (\forall d2 \in enjambreDrones(s)) |vueloSinRepeticiones(d1)| \leq |vueloSinRepeticiones(d2)|]$;

`aux vueloSinRepeticiones` (d:Drone) : [$\mathbb{Z}$, $\mathbb{Z}$] $= [vueloRealizado(d)_i \mid i \leftarrow [0... |vueloRealizado(d)|)(\forall j \in [0..i))\ vueloRealizado(d)_i \neq vueloRealizado(d)_j$;