TAD idPedido es int
TAD Pedido struct(id : idPedido, cant : int, dir : Direccion)
Tads AlgoConQueso

obs pedidosPendientes : conj(IdPedido)
obs pedidosPreparados : conj(IdPedido)
obs detallePedido : dict(IdPedido, Pedido)
obs pizzasHechas : int

`proc InitPizzeria` () : AlgoConQueso
`requiere` {true}
`asegura` $\{res.pedidosPendientes = \{\} \wedge res.pedidosPreparados = \{\} \wedge res.detallePedido = \{\} \wedge res.pizzasHechas = 0)\}$

`proc recibirPedido` (inout q: AlgoConQueso, in d : Direccion, in c:int)
`requiere` $\{c > 0 \wedge q_0 = q\}$
`asegura` $\{q.pedidosPendientes = q_0.pedidosPendiente \cup \{|q_0.detallePedido| + 1\}\}$
`asegura` $\{q.detallePedido = SetKey(q_0.detallePedido, |q_0.detallePedido| + 1, (|q_0.detallePedido| + 1, c, d)\}$
`asegura` $\{q.pedidosPreparados = q_0.pedidosPreparados \wedge q.pizzasHechas = q_0.pizzasHechas\}$

`proc terminarPedidos` (inout q : AlgoConQueso, in id : idPedido)
`requiere` $\{id \in q.pedidosPendientes \wedge q_0 = q\}$
`asegura` $\{q.pizzasHechas = q_0.pizzasHechas + 1\}$
´ `asegura` $\{q.pedidosPendientes = q_0.pedidosPendientes - \{id\} \wedge q.detallePedido = q_0.detallePedido\}$
`asegura` $\{(\forall x : \mathbb{Z})\ (x \in q_0.pedidosPendientes \wedge_L p_0.detallePedido[x].zona \neq p_0.detallePedido[id].zona) \longrightarrow_L$
$(\forall k : \mathbb{Z})\ (k \in q_0.pedidosPendientes \wedge_L q.detallePedido[x].zona \neq q.detallePedido[id].zona \leftrightarrow k \in q.pedidosPendientes)$
$\vee_L\ (\exists j : \mathbb{Z})\ (j \in q_0.pedidosPendientes \wedge_L q.detallePedido[j].zona = q.detallePedido[id].zona \wedge$
$q.pedidosPreparados = q_0.pedidosPreparados \cup \{id\})\}$

`proc maximoPedidoresPorZona` (in q : AlgoConQueso, in zona : int) : conj< Direccion >
`requiere` $\{(\exists i : \mathbb{Z})\ (i \in q.detallePedido \wedge_L q.detallePedido[i].zona = zona)\}$
`asegura` $\{(\forall j : res)\ ((\exists k : \mathbb{Z})\ (k \in q.detallePedido \wedge_L q.detallePedido[k].pedido.dir = j))\}$
`asegura` $\{(\forall i : \mathbb{Z})\ (i \in q.detallePedido \wedge_L q.detallePedido[i].zona = zona \longrightarrow_L$
$(\forall k : res)\ (q.detallePedidos[i] \leq k.cantidad))\}$

`proc cantidadPizzasHechas` (in q: AlgoConQueso) : $\mathbb{Z}$
`requiere` {true}
`asegura` $\{res = q.pizzasHechas\}$