

Honestidad académica y copias

Un trabajo práctico es un examen, por lo que debe realizarse de manera individual. La discusión y el intercambio de información de carácter general con los compañeros se permite (e incluso se recomienda), pero **NO A NIVEL DE CÓDIGO**. Igualmente el remitir código de terceros, **OBTENIDO A TRAVÉS DE LA RED** o cualquier otro medio, se considerará plagio.

Cualquier plagio o compartición de código que se detecte, significará automáticamente la calificación de **CERO EN LA ASIGNATURA** para **TODOS** los alumnos involucrados. Por tanto, a estos alumnos **NO** se les conservará, para futuras convocatorias, ninguna nota que hubiesen obtenido hasta el momento, **SIN PERJUICIO DE OTRAS MEDIDAS DE CARÁCTER DISCIPLINARIO QUE SE PUDIERAN TOMAR**.

ACL2: Trozos de listas

Una lista l_1 es un prefijo de otra lista l_2 , si todos los elementos de l_1 aparecen en el mismo orden y consecutivos al comienzo de l_2 . Por ejemplo, la lista '(1 2) es prefijo de '(1 2 3 4), pero no es prefijo de '(2 3 1 4) ni de '(2 1 3 4). Este concepto se formaliza en ACL2 mediante la siguiente función:

```
(defun prefijo (l1 l2)
  (cond ((endp l1) (equal l1 nil))
        ((endp l2) nil)
        ((equal (car l1) (car l2)) (prefijo (cdr l1) (cdr l2)))
        (t nil)))
```

Una lista l_1 es un trozo de otra lista l_2 , si todos los elementos de l_1 aparecen en el mismo orden y consecutivos en algún punto de l_2 . Por ejemplo, la lista '(2 3) es un trozo de '(1 2 3 4 5), pero no es prefijo de '(3 1 2 4) ni de '(1 3 2 4). Este concepto se formaliza en ACL2 mediante la siguiente función:

```
(defun trozo (l1 l2)
  (cond ((endp l2) (equal l1 nil))
        (t (or (prefijo l1 l2) (trozo l1 (cdr l2))))))
```

En estas condiciones se tiene que la relación trozo es transitiva y estable con respecto a la concatenación:

```
(defthm trozo-transitiva
  (implies (and (trozo l1 l2)
                (trozo l2 l3))
            (trozo l1 l3)))

(defthm trozo-append-1
  (implies (trozo l1 l2)
            (trozo l1 (append l2 l3))))

(defthm trozo-append-2
  (implies (trozo l1 l2)
            (trozo l1 (append l3 l2))))
```

También se tiene que si la lista l_1 es un trozo de la lista l_2 , y l_2 es un trozo de l_1 , entonces ambas listas son iguales:

```
(defthm trozo-equal
  (implies (and (trozo l1 l2)
                (trozo l2 l1))
            (equal l1 l2))
  :rule-classes nil)
```

La directiva `:rule-classes nil` indica al sistema ACL2 que la regla no debe usarse como regla de reescritura y no afecta para nada al proceso de demostración de la propiedad.

Se pide demostrar estos cuatro resultados en ACL2.

Evaluación

- Valoración del trabajo: **5** puntos.
- Los criterios de evaluación serán los siguientes:
 - Corrección de la solución.
 - Justificación de los resultados auxiliares necesitados.
 - Documentación del trabajo.
 - Presentación del trabajo (si se estima necesario).
- Plazo y forma de entrega:
 - El trabajo se desarrollará en un fichero admisible como fichero de entrada para el sistema ACL2, con el desarrollo correspondiente a la demostración de cada una de las cuatro propiedades. La documentación del trabajo se incluirá en dicho fichero como comentario.
 - La entrega se realizará vía Web a través de la página de la asignatura en el plazo habilitado para ello para la convocatoria en vigor. Dicho plazo se publicará en la página de la asignatura.