

#|

Copyright (C) 1994 by Computational Logic, Inc. All Rights Reserved.

This script is hereby placed in the public domain, and therefore unlimited editing and redistribution is permitted.

NO WARRANTY

Computational Logic, Inc. PROVIDES ABSOLUTELY NO WARRANTY. THE EVENT SCRIPT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE SCRIPT IS WITH YOU. SHOULD THE SCRIPT PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

IN NO EVENT WILL Computational Logic, Inc. BE LIABLE TO YOU FOR ANY DAMAGES, ANY LOST PROFITS, LOST MONIES, OR OTHER SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THIS SCRIPT (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY THIRD PARTIES), EVEN IF YOU HAVE ADVISED US OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY ANY OTHER PARTY.

|#

; William R. Bevier

EVENT: Start with the initial **nqthm** theory.

DEFINITION:

```
delete(x, l)
= if listp(l)
  then if x = car(l) then cdr(l)
        else cons(car(l), delete(x, cdr(l))) endif
  else l endif
```

DEFINITION:

```
bagdiff(x, y)
= if listp(y)
  then if car(y) ∈ x then bagdiff(delete(car(y), x), cdr(y))
        else bagdiff(x, cdr(y)) endif
  else x endif
```

DEFINITION:

```

bagint (x, y)
= if listp (x)
  then if car (x) ∈ y
    then cons (car (x), bagint (cdr (x), delete (car (x), y)))
    else bagint (cdr (x), y) endif
  else nil endif

```

DEFINITION:

```

occurrences (x, l)
= if listp (l)
  then if x = car (l) then 1 + occurrences (x, cdr (l))
    else occurrences (x, cdr (l)) endif
  else 0 endif

```

DEFINITION:

```

subbagp (x, y)
= if listp (x)
  then if car (x) ∈ y then subbagp (cdr (x), delete (car (x), y))
    else f endif
  else t endif

```

THEOREM: listp-delete

```

listp (delete (x, l))
= if listp (l) then (x ≠ car (l)) ∨ listp (cdr (l))
  else f endif

```

EVENT: Disable listp-delete.

THEOREM: delete-non-member

```

(x ∉ y) → (delete (x, y) = y)

```

THEOREM: delete-delete

```

delete (y, delete (x, z)) = delete (x, delete (y, z))

```

THEOREM: equal-occurrences-zero

```

(occurrences (x, l) = 0) = (x ∉ l)

```

THEOREM: member-non-list

```

(¬ listp (l)) → (x ∉ l)

```

THEOREM: member-delete

```

(x ∈ delete (y, l))
= if x ∈ l
  then if x = y then 1 < occurrences (x, l)
    else t endif
  else f endif

```

THEOREM: member-delete-implies-membership

$$(x \in \text{delete}(y, l)) \rightarrow (x \in l)$$

THEOREM: occurrences-delete

$$\begin{aligned} & \text{occurrences}(x, \text{delete}(y, l)) \\ = & \text{if } x = y \\ & \text{then if } x \in l \text{ then occurrences}(x, l) - 1 \\ & \text{else 0 endif} \\ & \text{else occurrences}(x, l) \text{ endif} \end{aligned}$$

THEOREM: member-bagdiff

$$(x \in \text{bagdiff}(a, b)) = (\text{occurrences}(x, b) < \text{occurrences}(x, a))$$

THEOREM: bagdiff-delete

$$\text{bagdiff}(\text{delete}(e, x), y) = \text{delete}(e, \text{bagdiff}(x, y))$$

THEOREM: subbagp-delete

$$\text{subbagp}(x, \text{delete}(u, y)) \rightarrow \text{subbagp}(x, y)$$

THEOREM: subbagp-cdr1

$$\text{subbagp}(x, y) \rightarrow \text{subbagp}(\text{cdr}(x), y)$$

THEOREM: subbagp-cdr2

$$\text{subbagp}(x, \text{cdr}(y)) \rightarrow \text{subbagp}(x, y)$$

THEOREM: subbagp-bagint1

$$\text{subbagp}(\text{bagint}(x, y), x)$$

THEOREM: subbagp-bagint2

$$\text{subbagp}(\text{bagint}(x, y), y)$$

THEOREM: occurrences-bagint

$$\begin{aligned} & \text{occurrences}(x, \text{bagint}(a, b)) \\ = & \text{if occurrences}(x, a) < \text{occurrences}(x, b) \text{ then occurrences}(x, a) \\ & \text{else occurrences}(x, b) \text{ endif} \end{aligned}$$

THEOREM: occurrences-bagdiff

$$\text{occurrences}(x, \text{bagdiff}(a, b)) = (\text{occurrences}(x, a) - \text{occurrences}(x, b))$$

THEOREM: member-bagint

$$(x \in \text{bagint}(a, b)) = ((x \in a) \wedge (x \in b))$$

EVENT: Let us define the theory *bags* to consist of the following events: occurrences-bagint, bagdiff-delete, occurrences-bagdiff, member-bagint, member-bagdiff, subbagp-bagint2, subbagp-bagint1, subbagp-cdr2, subbagp-cdr1, subbagp-delete.

EVENT: Make the library "bags" and compile it.

Index

bagdiff, 1, 3
bagdiff-delete, 3
bagint, 1–3
bags, 3

delete, 1–3
delete-delete, 2
delete-non-member, 2

equal-occurrences-zero, 2

listp-delete, 2

member-bagdiff, 3
member-bagint, 3
member-delete, 2
member-delete-implies-membership,
3
member-non-list, 2

occurrences, 2, 3
occurrences-bagdiff, 3
occurrences-bagint, 3
occurrences-delete, 3

subbagp, 2, 3
subbagp-bagint1, 3
subbagp-bagint2, 3
subbagp-cdr1, 3
subbagp-cdr2, 3
subbagp-delete, 3