```
#|

 Copyright (C) 1994 by Yuan Yu.  All Rights Reserved.

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

 NO WARRANTY

 Yuan Yu 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 Yuan Yu 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.

|#
```

EVENT: Start with the library "mc20-2" using the compiled version.

```
; some general theorems.  They are used to establish properties of string
; functions.
```

THEOREM: ilessp-lessp
$((x \in \mathbf{N}) \land (y \in \mathbf{N})) \rightarrow (\text{ilessp}\,(x,\ y) = (x < y))$

THEOREM: idifference-numberp
$(\text{idifference}\,(x,\ y) \in \mathbf{N}) = (\neg\ \text{ilessp}\,(x,\ y))$

THEOREM: idifference-equal-0
$(\text{idifference}\,(x,\ y) = \mathtt{0}) = (\text{fix-int}\,(x) = \text{fix-int}\,(y))$

EVENT: Enable get-nth-0.

EVENT: Enable put-nth-0.

```
; the upper bound of strlen.
```

THEOREM: strlen-ub
$\text{stringp}\,(i,\ n,\ \mathit{lst}) \rightarrow (\text{strlen}\,(i,\ n,\ \mathit{lst}) < n)$

```
; the lower bound of strlen.
```

THEOREM: strlen-lb
$\text{strlen}\,(i,\ n,\ \mathit{lst}) \not< i$

THEOREM: strlen-01

$$
\begin{aligned}
&(\text{strlen}\,(i,\ \mathtt{0},\ \mathit{lst}) = i)\\
\wedge\quad &(\text{strlen}\,(i,\ \mathtt{1},\ \mathit{lst})\\
&\quad = \ \textbf{if}\ (i < \mathtt{1}) \wedge (\text{get-nth}\,(\mathtt{0},\ \mathit{lst}) \neq \mathtt{0})\ \ \textbf{then}\ \mathtt{1}\\
&\qquad\quad \textbf{else}\ i\ \textbf{endif})\\
\wedge\quad &(\text{strlen}\,(\mathtt{1},\ n,\ \mathit{lst}) \neq \mathtt{0})
\end{aligned}
$$

THEOREM: strlen-ii
$\text{strlen}\,(i,\ i,\ \mathit{lst}) = i$

THEOREM: strlen-ii1

$$
\begin{aligned}
&\text{strlen}\,(i,\ 1 + i,\ \mathit{lst})\\
=\quad &\textbf{if}\ \text{get-nth}\,(i,\ \mathit{lst}) = \mathtt{0}\ \ \textbf{then}\ i\\
&\textbf{else}\ 1 + i\ \textbf{endif}
\end{aligned}
$$

```
; some properties of strlen.
; lst[j] =\= 0, if i <= j < strlen(lst).
```

THEOREM: strlen-non0p
$((i \leq j) \wedge (j < \text{strlen}\,(i,\ n,\ \mathit{lst}))) \rightarrow (\text{get-nth}\,(j,\ \mathit{lst}) \neq \mathtt{0})$

```
; lst[strlen] == 0!
```

THEOREM: strlen-0p
$\text{stringp}\,(i,\ n,\ \mathit{lst}) \rightarrow (\text{get-nth}\,(\text{strlen}\,(i,\ n,\ \mathit{lst}),\ \mathit{lst}) = \mathtt{0})$

```
; some properties of strcpy.
```

THEOREM: strcpy-get-1
$(j < i) \rightarrow (\text{get-nth}\,(j,\ \text{strcpy}\,(i,\ \mathit{lst1},\ \mathit{n2},\ \mathit{lst2})) = \text{get-nth}\,(j,\ \mathit{lst1}))$

```
; strlen(strcpy(lst1, lst2)) == strlen(lst2).
```

THEOREM: strcpy-strlen
$\text{strlen}\,(i,\ \mathit{n2},\ \text{strcpy}\,(i,\ \mathit{lst1},\ \mathit{n2},\ \mathit{lst2})) = \text{strlen}\,(i,\ \mathit{n2},\ \mathit{lst2})$

```
;  if i <= j <= strlen(lst2), lst1[j] == lst2[j].
```

THEOREM: strcpy-cpy

$(\mathrm{stringp}\,(i,\ n2,\ lst2) \land (i \leq j) \land (j \leq \mathrm{strlen}\,(i,\ n2,\ lst2)))$
$\rightarrow\quad (\text{get-nth}\,(j,\ \mathrm{strcpy}\,(i,\ lst1,\ n2,\ lst2)) = \text{get-nth}\,(j,\ lst2))$

```
;  if lst2 is a string, then lst1' is also a string.
```

THEOREM: strcpy-stringp

$\mathrm{stringp}\,(i,\ n2,\ lst2) \rightarrow \mathrm{stringp}\,(i,\ n2,\ \mathrm{strcpy}\,(i,\ lst1,\ n2,\ lst2))$

```
; some properties of strcmp.
; (strcmp lst lst) == 0.
```

THEOREM: strcmp-id

$\mathrm{strcmp}\,(i,\ n,\ lst,\ lst) = 0$

```
; (strcmp lst1 lst2) < 0 ==> (strcmp lst2 lst1) >= 0.
```

THEOREM: strcmp-antisym

$\mathrm{negativep}\,(\mathrm{strcmp}\,(i,\ n,\ lst1,\ lst2)) = (0 < \mathrm{strcmp}\,(i,\ n,\ lst2,\ lst1))$

```
; the transitivity of strcmp.
```

THEOREM: strcmp-trans-1

$(\text{lst-of-chrp}\,(lst1)$
$\land\quad \text{lst-of-chrp}\,(lst2)$
$\land\quad \text{lst-of-chrp}\,(lst3)$
$\land\quad (\mathrm{strcmp}\,(i,\ n,\ lst1,\ lst2) \in \mathbf{N})$
$\land\quad (\mathrm{strcmp}\,(i,\ n,\ lst2,\ lst3) \in \mathbf{N}))$
$\rightarrow\quad (\mathrm{strcmp}\,(i,\ n,\ lst1,\ lst3) \in \mathbf{N})$

```
; (strcpy lst1 lst2) == lst2.
```

THEOREM: strcmp-strcpy

$\mathrm{strcmp}\,(i,\ n,\ lst2,\ \mathrm{strcpy}\,(i,\ lst1,\ n,\ lst2)) = 0$

```
; if (strcmp lst1 lst2) == 0, all i <= j < (strlen lst1) lst1[j] == lst2[j].
```

THEOREM: strcmp-thm1

$(\text{lst-of-chrp}\,(lst1)$
$\land\quad \text{lst-of-chrp}\,(lst2)$
$\land\quad (\mathrm{strcmp}\,(i,\ n,\ lst1,\ lst2) = 0)$
$\land\quad (i \leq j)$
$\land\quad (j < \mathrm{strlen}\,(i,\ n,\ lst1)))$
$\rightarrow\quad (\text{get-nth}\,(j,\ lst1) = \text{get-nth}\,(j,\ lst2))$

DEFINITION:
strcmp-sk $(n1,\ lst1,\ lst2)$
$$\leftrightarrow \quad \exists\, j\ (\forall\, i\ ((i < j) \rightarrow (\text{get-nth}\,(i,\ lst1) = \text{get-nth}\,(i,\ lst2))) \wedge (\text{strcmp}\,(0,\ n1,\ lst1,\ lst2) = \text{idifference}\,(\text{get-nth}\,(j,\ lst1),\ \text{get-nth}\,(j,\ lst2))))$$

DEFINITION:
strcmp-j $(i,\ n1,\ lst1,\ lst2)$
= **if** $i < n1$
then if get-nth $(i,\ lst1)$ = get-nth $(i,\ lst2)$
then if get-nth $(i,\ lst1)$ = NULL **then** fix (i)
else strcmp-j $(1 + i,\ n1,\ lst1,\ lst2)$ **endif**
else fix (i) **endif**
else fix (i) **endif**

THEOREM: strcmp-j-1
$$((i < \text{strcmp-j}\,(i1,\ n1,\ lst1,\ lst2)) \wedge (i1 \leq i)) \rightarrow (\text{get-nth}\,(i,\ lst1) = \text{get-nth}\,(i,\ lst2))$$

THEOREM: strcmp-j-2
$$(\text{strcmp}\,(i1,\ n1,\ lst1,\ lst2) \neq 0) \rightarrow (\text{strcmp}\,(i1,\ n1,\ lst1,\ lst2) = \text{idifference}\,(\text{get-nth}\,(\text{strcmp-j}\,(i1,\ n1,\ lst1,\ lst2),\ lst1),\ \text{get-nth}\,(\text{strcmp-j}\,(i1,\ n1,\ lst1,\ lst2),\ lst2)))$$

THEOREM: strcmp-thm2
$$(\text{strcmp}\,(0,\ n1,\ lst1,\ lst2) \neq 0) \rightarrow \text{strcmp-sk}\,(n1,\ lst1,\ lst2)$$

EVENT: Disable strcmp-j-1.

EVENT: Disable strcmp-j-2.

```
; some properties of strcat.
```

THEOREM: strcpy1-get-1
$$(j < i1) \rightarrow (\text{get-nth}\,(j,\ \text{strcpy1}\,(i1,\ lst1,\ i2,\ n2,\ lst2)) = \text{get-nth}\,(j,\ lst1))$$

DEFINITION:
strlen1 $(i,\ n,\ lst)$
= **if** $i < n$
then if get-nth $(i,\ lst)$ = NULL **then** 0
else $1 +$ strlen1 $(1 + i,\ n,\ lst)$ **endif**
else 0 **endif**

THEOREM: strlen1-strlen
$\text{strlen1}(i,\ n,\ lst) = (\text{strlen}(i,\ n,\ lst) - i)$

EVENT: Disable strlen1-strlen.

THEOREM: strcpy1-get-2
$((i1 + \text{strlen1}(i2,\ n2,\ lst2)) < j)$
$\rightarrow\ (\text{get-nth}(j,\ \text{strcpy1}(i1,\ lst1,\ i2,\ n2,\ lst2)) = \text{get-nth}(j,\ lst1))$

THEOREM: get-nth-plus-0
$(j \notin \mathbf{N}) \rightarrow (\text{get-nth}(i + j,\ lst) = \text{get-nth}(i,\ lst))$

THEOREM: plus-sub1-add1
$((x + (1 + y)) - 1) = (x + y)$

THEOREM: strcpy1-get-3
$((i1 \leq j) \land (j < (i1 + \text{strlen1}(i2,\ n2,\ lst2))))$
$\rightarrow\ (\text{get-nth}(j,\ \text{strcpy1}(i1,\ lst1,\ i2,\ n2,\ lst2))$
$\quad = \text{get-nth}(i2 + (j - i1),\ lst2))$

```
; all 0 <= j < (strlen lst1) lst1'[j] == lst1[j].
```

THEOREM: strcat-get-1
$(j < \text{strlen}(\mathtt{0},\ n1,\ lst1))$
$\rightarrow\ (\text{get-nth}(j,\ \text{strcat}(n1,\ lst1,\ n2,\ lst2)) = \text{get-nth}(j,\ lst1))$

```
; all (strlen lst1) <= j < (strlen lst1)+(strlen lst2) lst1'[j] == lst2[j].
```

THEOREM: strcat-get-2
$(\text{stringp}(\mathtt{0},\ n1,\ lst1)$
$\land\ (\text{strlen}(\mathtt{0},\ n1,\ lst1) \leq j)$
$\land\ (j < (\text{strlen}(\mathtt{0},\ n1,\ lst1) + \text{strlen}(\mathtt{0},\ n2,\ lst2))))$
$\rightarrow\ (\text{get-nth}(j,\ \text{strcat}(n1,\ lst1,\ n2,\ lst2))$
$\quad = \text{get-nth}(j - \text{strlen}(\mathtt{0},\ n1,\ lst1),\ lst2))$

```
; some properties about strchr.
; lst[strchr] == ch,  if strchr returns non-f.
```

THEOREM: strchr-thm1
$\text{strchr}(i,\ n,\ lst,\ ch) \rightarrow (\text{get-nth}(\text{strchr}(i,\ n,\ lst,\ ch),\ lst) = ch)$

```
; all i <= j < strchr, lst[j] =\= ch.  i.e. strchr is the first one.
```

THEOREM: strchr-thm2
$((i \leq j) \land (j < \text{strchr}(i,\ n,\ lst,\ ch))) \rightarrow (\text{get-nth}(j,\ lst) \neq ch)$

```
; all i <= j < strlen, lst[j] =\= ch, if strchr returns f.
```

THEOREM: strchr-thm3
$((\neg\ \text{strchr}\,(i,\ n,\ \mathit{lst},\ \mathit{ch})) \land (i \leq j) \land (j < \text{strlen}\,(i,\ n,\ \mathit{lst})))$
$\rightarrow \quad (\text{get-nth}\,(j,\ \mathit{lst}) \neq \mathit{ch})$

```
; ch is not equal to the null character, if strchr returns f.
```

THEOREM: strchr-thm4
$(\text{stringp}\,(i,\ n,\ \mathit{lst}) \land (\neg\ \text{strchr}\,(i,\ n,\ \mathit{lst},\ \mathit{ch}))) \rightarrow (\mathit{ch} \neq \mathtt{0})$

```
; strcpy and strchr.
```

THEOREM: strcpy-strchr
$\text{strchr}\,(i,\ n,\ \text{strcpy}\,(i,\ \mathit{lst1},\ n,\ \mathit{lst2}),\ \mathit{ch}) = \text{strchr}\,(i,\ n,\ \mathit{lst2},\ \mathit{ch})$

```
; some properties about memset.
; lemmas about memset1.
```

THEOREM: memset1-get-1
$(j < i) \rightarrow (\text{get-nth}\,(j,\ \text{memset1}\,(i,\ n,\ \mathit{lst},\ \mathit{ch})) = \text{get-nth}\,(j,\ \mathit{lst}))$

```
; all i <= j < i+n, lst'[j] == ch.
```

THEOREM: memset1-thm1
$((i \leq j) \land (j < (i + n)))$
$\rightarrow \quad (\text{get-nth}\,(j,\ \text{memset1}\,(i,\ n,\ \mathit{lst},\ \mathit{ch})) = \mathit{ch})$

THEOREM: memset1-thm2
$(((i + n) \leq j) \land (n \not\simeq \mathtt{0}))$
$\rightarrow \quad (\text{get-nth}\,(j,\ \text{memset1}\,(i,\ n,\ \mathit{lst},\ \mathit{ch})) = \text{get-nth}\,(j,\ \mathit{lst}))$

```
; all i <= j < n, lst'[j] == ch.
```

THEOREM: memset-thm1
$((i \leq j) \land (j < n)) \rightarrow (\text{get-nth}\,(j,\ \text{memset}\,(n,\ \mathit{lst},\ \mathit{ch})) = \mathit{ch})$

```
; all j >= n, lst'[j] == lst[j].
```

THEOREM: memset-thm2
$((n \leq j) \land (n \in \mathbf{N}))$
$\rightarrow \quad (\text{get-nth}\,(j,\ \text{memset}\,(n,\ \mathit{lst},\ \mathit{ch})) = \text{get-nth}\,(j,\ \mathit{lst}))$

```
; some properties about memchr.
```

THEOREM: memchr1-thm1
$\text{memchr1}\,(i,\ n,\ \mathit{lst},\ \mathit{ch}) \rightarrow (\text{get-nth}\,(\text{memchr1}\,(i,\ n,\ \mathit{lst},\ \mathit{ch}),\ \mathit{lst}) = \mathit{ch})$

```
; lst[memchr] == ch,  if memchr returns non-f.
```

THEOREM: memchr-thm1
$\text{memchr}\,(n,\ \mathit{lst},\ \mathit{ch}) \rightarrow (\text{get-nth}\,(\text{memchr}\,(n,\ \mathit{lst},\ \mathit{ch}),\ \mathit{lst}) = \mathit{ch})$

THEOREM: memchr1-thm2
$((i \leq j) \wedge (j < \text{memchr1}\,(i,\ n,\ \mathit{lst},\ \mathit{ch}))) \rightarrow (\text{get-nth}\,(j,\ \mathit{lst}) \neq \mathit{ch})$

```
; all j < memchr, lst[j] =\= ch.  i.e. memchr is the first one.
```

THEOREM: memchr-thm2
$(j < \text{memchr}\,(n,\ \mathit{lst},\ \mathit{ch})) \rightarrow (\text{get-nth}\,(j,\ \mathit{lst}) \neq \mathit{ch})$

THEOREM: memchr1-thm3
$((\neg\ \text{memchr1}\,(i,\ n,\ \mathit{lst},\ \mathit{ch})) \wedge (i \leq j) \wedge (j < (i + n)))$
$\rightarrow \quad (\text{get-nth}\,(j,\ \mathit{lst}) \neq \mathit{ch})$

```
; all j < n, lst[j] =\= ch, if memchr returns f.
```

THEOREM: memchr-thm3
$((\neg\ \text{memchr}\,(n,\ \mathit{lst},\ \mathit{ch})) \wedge (j < n)) \rightarrow (\text{get-nth}\,(j,\ \mathit{lst}) \neq \mathit{ch})$

```
; some properties about strrchr.
```

THEOREM: strrchr-la1
$(j \in \mathbf{N}) \rightarrow \text{strrchr}\,(i,\ n,\ \mathit{lst},\ \mathit{ch},\ j)$

THEOREM: strrchr-strchr-la
$(\neg\ \text{strchr}\,(i,\ n,\ \mathit{lst},\ \mathit{ch})) \rightarrow (\text{strrchr}\,(i,\ n,\ \mathit{lst},\ \mathit{ch},\ j) = j)$

THEOREM: strrchr-la2
$\text{strchr}\,(i,\ n,\ \mathit{lst},\ \mathit{ch}) \rightarrow (\text{get-nth}\,(\text{strrchr}\,(i,\ n,\ \mathit{lst},\ \mathit{ch},\ j),\ \mathit{lst}) = \mathit{ch})$

```
; strrchr finds ch in the string.
```

THEOREM: strrchr-thm1
$\text{strrchr}\,(i,\ n,\ \mathit{lst},\ \mathit{ch},\ \mathbf{f}) \rightarrow (\text{get-nth}\,(\text{strrchr}\,(i,\ n,\ \mathit{lst},\ \mathit{ch},\ \mathbf{f}),\ \mathit{lst}) = \mathit{ch})$

THEOREM: strchr-get
$((i \leq j) \wedge (j < \text{strlen}\,(i,\ n,\ \mathit{lst}))) \rightarrow \text{strchr}\,(i,\ n,\ \mathit{lst},\ \text{get-nth}\,(j,\ \mathit{lst}))$

THEOREM: strrchr-la3
$(\text{strchr}\,(i,\ n,\ \mathit{lst},\ \mathit{ch})$
$\wedge \quad (\text{strrchr}\,(i,\ n,\ \mathit{lst},\ \mathit{ch},\ k) < j)$
$\wedge \quad (j < \text{strlen}\,(i,\ n,\ \mathit{lst})))$
$\rightarrow \quad (\text{get-nth}\,(j,\ \mathit{lst}) \neq \mathit{ch})$

```
; strrchr returns the last one.
```

THEOREM: strrchr-thm2

$$
\begin{array}{ll}
\multicolumn{2}{l}{(\mathrm{strrchr}\,(i,\ n,\ lst,\ ch,\ \mathbf{f})} \\
\ \wedge & (\mathrm{strrchr}\,(i,\ n,\ lst,\ ch,\ \mathbf{f}) < j) \\
\ \wedge & (j < \mathrm{strlen}\,(i,\ n,\ lst))) \\
\rightarrow & (\text{get-nth}\,(j,\ lst) \neq ch)
\end{array}
$$

```
; no means none.
```

THEOREM: strrchr-thm3

$$
\begin{array}{ll}
\multicolumn{2}{l}{((\neg\ \mathrm{strrchr}\,(i,\ n,\ lst,\ ch,\ k)) \wedge (i \le j) \wedge (j < \mathrm{strlen}\,(i,\ n,\ lst)))} \\
\rightarrow & (\text{get-nth}\,(j,\ lst) \neq ch)
\end{array}
$$

```
; ch is not equal to the null character, if strrchr returns f.
```

THEOREM: strrchr-thm4

$$(\mathrm{stringp}\,(i,\ n,\ lst) \wedge (\neg\ \mathrm{strrchr}\,(i,\ n,\ lst,\ ch,\ k))) \rightarrow (ch \neq \mathtt{0})$$

```
; some properties about memcmp.
```

THEOREM: memcmp1-id

$$\mathrm{memcmp1}\,(i,\ n,\ lst,\ lst) = \mathtt{0}$$

THEOREM: memcmp-id

$$\mathrm{memcmp}\,(n,\ lst,\ lst) = \mathtt{0}$$

THEOREM: memcmp1-thm1

$$
\begin{array}{ll}
\multicolumn{2}{l}{(\text{lst-of-chrp}\,(lst1)} \\
\ \wedge & \text{lst-of-chrp}\,(lst2) \\
\ \wedge & (\mathrm{memcmp1}\,(i,\ n,\ lst1,\ lst2) = \mathtt{0}) \\
\ \wedge & (i \le j) \\
\ \wedge & (j < (i + n))) \\
\rightarrow & (\text{get-nth}\,(j,\ lst1) = \text{get-nth}\,(j,\ lst2))
\end{array}
$$

THEOREM: memcmp-thm1

$$
\begin{array}{ll}
\multicolumn{2}{l}{(\text{lst-of-chrp}\,(lst1)} \\
\ \wedge & \text{lst-of-chrp}\,(lst2) \\
\ \wedge & (\mathrm{memcmp}\,(n,\ lst1,\ lst2) = \mathtt{0}) \\
\ \wedge & (j < n)) \\
\rightarrow & (\text{get-nth}\,(j,\ lst1) = \text{get-nth}\,(j,\ lst2))
\end{array}
$$

DEFINITION:

$$
\begin{array}{l}
\text{memcmp-sk}\,(n,\ lst1,\ lst2) \\
\leftrightarrow\quad \exists\, j\ (\forall\, i\ ((i < j) \rightarrow (\text{get-nth}\,(i,\ lst1) = \text{get-nth}\,(i,\ lst2))) \\
\qquad\qquad \wedge\quad (\mathrm{memcmp}\,(n,\ lst1,\ lst2) \\
\qquad\qquad\qquad =\quad \mathrm{idifference}\,(\text{get-nth}\,(j,\ lst1),\ \text{get-nth}\,(j,\ lst2))))
\end{array}
$$

DEFINITION:
memcmp-j $(i,\ n,\ \mathit{lst1},\ \mathit{lst2})$
$=$ **if** get-nth $(i,\ \mathit{lst1})$ = get-nth $(i,\ \mathit{lst2})$
then if $(n - 1) = \mathtt{0}$ **then** fix (i)
else memcmp-j $(1 + i,\ n - 1,\ \mathit{lst1},\ \mathit{lst2})$ **endif**
else fix (i) **endif**

THEOREM: memcmp-j-1
$((i <$ memcmp-j $(\mathit{i1},\ \mathit{n1},\ \mathit{lst1},\ \mathit{lst2})) \land (\mathit{i1} \le i))$
$\rightarrow$ (get-nth $(i,\ \mathit{lst1})$ = get-nth $(i,\ \mathit{lst2})$)

THEOREM: memcmp-j-2
(memcmp1 $(\mathit{i1},\ \mathit{n1},\ \mathit{lst1},\ \mathit{lst2}) \neq \mathtt{0}$)
$\rightarrow$ (memcmp1 $(\mathit{i1},\ \mathit{n1},\ \mathit{lst1},\ \mathit{lst2})$
$=$ idifference (get-nth (memcmp-j $(\mathit{i1},\ \mathit{n1},\ \mathit{lst1},\ \mathit{lst2}),\ \mathit{lst1}$),
get-nth (memcmp-j $(\mathit{i1},\ \mathit{n1},\ \mathit{lst1},\ \mathit{lst2}),\ \mathit{lst2}$)))

THEOREM: memcmp-thm2
(memcmp $(n,\ \mathit{lst1},\ \mathit{lst2}) \neq \mathtt{0}$) $\rightarrow$ memcmp-sk $(n,\ \mathit{lst1},\ \mathit{lst2})$

EVENT: Disable memcmp-j-1.

EVENT: Disable memcmp-j-2.

```
; some properties about strncat.
```

THEOREM: strcpy2-get-1
$(j < \mathit{i1}) \rightarrow$ (get-nth $(j$, strcpy2 $(\mathit{i1},\ \mathit{lst1},\ \mathit{i2},\ \mathit{n2},\ \mathit{lst2}))$ = get-nth $(j,\ \mathit{lst1})$)

THEOREM: strlen1-ii
strlen1 $(i,\ i,\ \mathit{lst}) = \mathtt{0}$

THEOREM: strcpy2-get-2
$((\mathit{i1} \le j) \land (j < (\mathit{i1}$ + strlen1 $(\mathit{i2},\ \mathit{i2} + \mathit{n2},\ \mathit{lst2}))))$
$\rightarrow$ (get-nth $(j$, strcpy2 $(\mathit{i1},\ \mathit{lst1},\ \mathit{i2},\ \mathit{n2},\ \mathit{lst2}))$
$=$ get-nth $((\mathit{i2} + j) - \mathit{i1},\ \mathit{lst2})$)

```
; all j < (strlen lst1), lst'[j] == lst1[j].
```

THEOREM: strncat-get-1
$(j <$ strlen $(\mathtt{0},\ \mathit{n1},\ \mathit{lst1}))$
$\rightarrow$ (get-nth $(j$, strncat $(\mathit{n1},\ \mathit{lst1},\ \mathit{n2},\ \mathit{lst2}))$ = get-nth $(j,\ \mathit{lst1})$)

```
; all (strlen lst1) <= j < (strlen lst1)+(strlen lst2),
; lst'[j] == lst2[j-(strlen lst1)].
```

THEOREM: strncat-get-2
$(\text{stringp}(0, n1, lst1)$
$\land \quad (\text{strlen}(0, n1, lst1) \leq j)$
$\land \quad (j < (\text{strlen}(0, n1, lst1) + \text{strlen}(0, n2, lst2))))$
$\rightarrow \quad (\text{get-nth}(j, \text{strncat}(n1, lst1, n2, lst2))$
$\quad = \quad \text{get-nth}(j - \text{strlen}(0, n1, lst1), lst2))$

```
; some properties about strncpy.
```

THEOREM: zero-list1-la
$(j < i) \rightarrow (\text{get-nth}(j, \text{zero-list1}(i, n, lst)) = \text{get-nth}(j, lst))$

THEOREM: zero-list1-get
$((i \leq j) \land (j < ((i + n) - 1)))$
$\rightarrow \quad (\text{get-nth}(j, \text{zero-list1}(i, n, lst)) = 0)$

THEOREM: zero-list-get
$((i \leq j) \land (j < (i + n)))$
$\rightarrow \quad (\text{get-nth}(j, \text{zero-list}(i, n, lst)) = 0)$

THEOREM: strncpy1-get
$(j < i) \rightarrow (\text{get-nth}(j, \text{strncpy1}(i, n, lst1, lst2)) = \text{get-nth}(j, lst1))$

EVENT: Disable zero-list.

THEOREM: strncpy1-strlen
$\text{strlen}(i, i + n, \text{strncpy1}(i, n, lst1, lst2)) = \text{strlen}(i, i + n, lst2)$

THEOREM: strcpy-0
$\text{strncpy}(0, lst1, lst2) = lst1$

```
; the length of (strncpy lst1 lst2) equals the length of lst2.
```

THEOREM: strncpy-strlen
$\text{strlen}(0, n, \text{strncpy}(n, lst1, lst2))$
$= \quad \textbf{if } n \simeq 0 \textbf{ then } \text{strlen}(0, n, lst1)$
$\quad \textbf{else } \text{strlen}(0, n, lst2) \textbf{ endif}$

THEOREM: strncpy1-cpy
$((i \leq j) \land (j < \text{strlen}(i, i + n, lst2)))$
$\rightarrow \quad (\text{get-nth}(j, \text{strncpy1}(i, n, lst1, lst2)) = \text{get-nth}(j, lst2))$

```
; all j < (strlen lst2), lst'[j] == lst2[j].
```

THEOREM: strncpy-cpy
$(j < \text{strlen}(0, n, lst2))$
$\rightarrow \quad (\text{get-nth}(j, \text{strncpy}(n, lst1, lst2)) = \text{get-nth}(j, lst2))$

THEOREM: strncpy1-0s
$((\text{strlen}(i,\ i + n,\ \mathit{lst2}) \le j) \land (j < (i + n)))$
$\rightarrow \quad (\text{get-nth}(j,\ \text{strncpy1}(i,\ n,\ \mathit{lst1},\ \mathit{lst2})) = \mathtt{0})$

```
; all (strlen lst2) <= j < n, lst'[j] == 0.
```

THEOREM: strncpy-0s
$((\text{strlen}(\mathtt{0},\ n,\ \mathit{lst2}) \le j) \land (j < n))$
$\rightarrow \quad (\text{get-nth}(j,\ \text{strncpy}(n,\ \mathit{lst1},\ \mathit{lst2})) = \mathtt{0})$

```
; some properties about strncmp.
```

THEOREM: strncmp1-id
$\text{strncmp1}(i,\ n,\ \mathit{lst},\ \mathit{lst}) = \mathtt{0}$

THEOREM: strncmp-id
$\text{strncmp}(n,\ \mathit{lst},\ \mathit{lst}) = \mathtt{0}$

THEOREM: strncmp1-thm1
$(\text{lst-of-chrp}(\mathit{lst1})$
$\land \quad \text{lst-of-chrp}(\mathit{lst2})$
$\land \quad (\text{strncmp1}(i,\ n,\ \mathit{lst1},\ \mathit{lst2}) = \mathtt{0})$
$\land \quad (i \le j)$
$\land \quad (j < \text{strlen}(i,\ i + n,\ \mathit{lst1})))$
$\rightarrow \quad (\text{get-nth}(j,\ \mathit{lst1}) = \text{get-nth}(j,\ \mathit{lst2}))$

THEOREM: strncmp-thm1
$(\text{lst-of-chrp}(\mathit{lst1})$
$\land \quad \text{lst-of-chrp}(\mathit{lst2})$
$\land \quad (\text{strncmp}(n,\ \mathit{lst1},\ \mathit{lst2}) = \mathtt{0})$
$\land \quad (j < \text{strlen}(\mathtt{0},\ n,\ \mathit{lst1})))$
$\rightarrow \quad (\text{get-nth}(j,\ \mathit{lst1}) = \text{get-nth}(j,\ \mathit{lst2}))$

DEFINITION:
$\text{strncmp-j}(i,\ n,\ \mathit{lst1},\ \mathit{lst2})$
$= \quad$ **if** $\text{get-nth}(i,\ \mathit{lst1}) = \text{get-nth}(i,\ \mathit{lst2})$
then if $\text{get-nth}(i,\ \mathit{lst1}) = \mathtt{0}$ **then** $\text{fix}(i)$
elseif $(n - 1) = \mathtt{0}$ **then** $\text{fix}(i)$
else $\text{strncmp-j}(1 + i,\ n - 1,\ \mathit{lst1},\ \mathit{lst2})$ **endif**
else $\text{fix}(i)$ **endif**

DEFINITION:
$\text{strncmp-sk}(n,\ \mathit{lst1},\ \mathit{lst2})$
$\leftrightarrow \quad \exists\, j\ (\forall\, i\ ((i < j) \rightarrow (\text{get-nth}(i,\ \mathit{lst1}) = \text{get-nth}(i,\ \mathit{lst2})))$
$\land \quad (\text{strncmp}(n,\ \mathit{lst1},\ \mathit{lst2})$
$= \quad \text{idifference}(\text{get-nth}(j,\ \mathit{lst1}),\ \text{get-nth}(j,\ \mathit{lst2}))))$

THEOREM: strncmp-j-1
$((i < \text{strncmp-j}\,(i1,\ n1,\ lst1,\ lst2)) \land (i1 \le i))$
$\rightarrow \quad (\text{get-nth}\,(i,\ lst1) = \text{get-nth}\,(i,\ lst2))$

THEOREM: strncmp-j-2
$(\text{strncmp1}\,(i1,\ n1,\ lst1,\ lst2) \neq \texttt{0})$
$\rightarrow \quad (\text{strncmp1}\,(i1,\ n1,\ lst1,\ lst2)$
$\quad = \quad \text{idifference}\,(\text{get-nth}\,(\text{strncmp-j}\,(i1,\ n1,\ lst1,\ lst2),\ lst1),$
$\qquad \text{get-nth}\,(\text{strncmp-j}\,(i1,\ n1,\ lst1,\ lst2),\ lst2)))$

THEOREM: strncmp-thm2
$(\text{strncmp}\,(n,\ lst1,\ lst2) \neq \texttt{0}) \rightarrow \text{strncmp-sk}\,(n,\ lst1,\ lst2)$

EVENT: Disable strncmp-j-1.

EVENT: Disable strncmp-j-2.

```
; some properties about strpbrk.
```

THEOREM: strpbrk-thm1
let j **be** $\text{strpbrk}\,(i1,\ n1,\ lst1,\ n2,\ lst2)$
in
$j \rightarrow \text{strchr1}\,(\texttt{0},\ n2,\ lst2,\ \text{get-nth}\,(j,\ lst1))$ **endlet**

THEOREM: strpbrk-thm2
$((i1 \le j) \land (j < \text{strpbrk}\,(i1,\ n1,\ lst1,\ n2,\ lst2)))$
$\rightarrow \quad (\neg\ \text{strchr1}\,(\texttt{0},\ n2,\ lst2,\ \text{get-nth}\,(j,\ lst1)))$

THEOREM: strpbrk-thm3
$((\neg\ \text{strpbrk}\,(i1,\ n1,\ lst1,\ n2,\ lst2))$
$\land \quad (i1 \le j)$
$\land \quad (j < \text{strlen}\,(i1,\ n1,\ lst1)))$
$\rightarrow \quad (\neg\ \text{strchr1}\,(\texttt{0},\ n2,\ lst2,\ \text{get-nth}\,(j,\ lst1)))$

```
; strchr1, which is used to specify several string functions, is a variant
; of strchr.  strchr1 does not have the 'thm4' of strchr.
```

THEOREM: strchr1-thm1
$\text{strchr1}\,(i,\ n,\ lst,\ ch) \rightarrow (\text{get-nth}\,(\text{strchr1}\,(i,\ n,\ lst,\ ch),\ lst) = ch)$

THEOREM: strchr1-thm2
$((i \le j) \land (j < \text{strchr1}\,(i,\ n,\ lst,\ ch))) \rightarrow (\text{get-nth}\,(j,\ lst) \neq ch)$

THEOREM: strchr1-thm3
$((\neg\ \text{strchr1}\,(i,\ n,\ lst,\ ch)) \land (i \le j) \land (j < \text{strlen}\,(i,\ n,\ lst)))$
$\rightarrow \quad (\text{get-nth}\,(j,\ lst) \neq ch)$

```
; some properties of strcspn.
```

THEOREM: strcspn-thm1
let j **be** strcspn ($i1$, $n1$, $lst1$, $n2$, $lst2$)
in
$j \rightarrow$ strchr (0, $n2$, $lst2$, get-nth (j, $lst1$)) **endlet**

THEOREM: strcspn-thm2
$((i1 \le j) \wedge (j <$ strcspn ($i1$, $n1$, $lst1$, $n2$, $lst2$)))
$\rightarrow$ ($\neg$ strchr (0, $n2$, $lst2$, get-nth (j, $lst1$)))

THEOREM: strcspn-thm3
(($\neg$ strcspn ($i1$, $n1$, $lst1$, $n2$, $lst2$))
$\wedge$ $(i1 \le j)$
$\wedge$ $(j <$ strlen ($i1$, $n1$, $lst1$)))
$\rightarrow$ ($\neg$ strchr (0, $n2$, $lst2$, get-nth (j, $lst1$)))

```
; some properties of strspn.
```

THEOREM: strspn-thm1
let j **be** strspn ($i1$, $n1$, $lst1$, $n2$, $lst2$)
in
$j \rightarrow$ ($\neg$ strchr1 (0, $n2$, $lst2$, get-nth (j, $lst1$))) **endlet**

THEOREM: strspn-thm2
$((i1 \le j) \wedge (j <$ strspn ($i1$, $n1$, $lst1$, $n2$, $lst2$)))
$\rightarrow$ strchr1 (0, $n2$, $lst2$, get-nth (j, $lst1$))

THEOREM: strspn-thm3
(($\neg$ strspn ($i1$, $n1$, $lst1$, $n2$, $lst2$))
$\wedge$ $(i1 \le j)$
$\wedge$ $(j <$ strlen ($i1$, $n1$, $lst1$)))
$\rightarrow$ strchr1 (0, $n2$, $lst2$, get-nth (j, $lst1$))

```
; a generalized defn for strncmp.
```

DEFINITION:
strncmp2 (i, n, $lst1$, j, $lst2$)
= **if** $n \simeq 0$ **then** 0
elseif get-nth (i, $lst1$) = get-nth (j, $lst2$)
then if get-nth (i, $lst1$) = 0 **then** 0
else strncmp2 ($1 + i$, $n - 1$, $lst1$, $1 + j$, $lst2$) **endif**
else idifference (get-nth (i, $lst1$), get-nth (j, $lst2$)) **endif**

THEOREM: strncmp2-non-numberp
$(i \notin \mathbf{N}) \rightarrow$ (strncmp2 (i, n, $lst1$, j, $lst2$) = strncmp2 (0, n, $lst1$, j, $lst2$))

THEOREM: strncmp2-0
$\text{strncmp2}\,(i,\ \mathtt{0},\ \mathit{lst1},\ j,\ \mathit{lst2}) = \mathtt{0}$

THEOREM: strncmp1-strncmp2
$(n \not\simeq \mathtt{0}) \rightarrow (\text{strncmp1}\,(i,\ n,\ \mathit{lst1},\ \mathit{lst2}) = \text{strncmp2}\,(i,\ n,\ \mathit{lst1},\ i,\ \mathit{lst2}))$

THEOREM: strncmp-strncmp2
$\text{strncmp}\,(n,\ \mathit{lst1},\ \mathit{lst2}) = \text{strncmp2}\,(\mathtt{0},\ n,\ \mathit{lst1},\ \mathtt{0},\ \mathit{lst2})$

THEOREM: strncmp2-mcdr-1
$\text{strncmp2}\,(\mathit{i1},\ n,\ \text{mcdr}\,(j,\ \mathit{lst1}),\ \mathit{i2},\ \mathit{lst2}) = \text{strncmp2}\,(\mathit{i1} + j,\ n,\ \mathit{lst1},\ \mathit{i2},\ \mathit{lst2})$

THEOREM: strncmp2-mcdr-2
$\text{strncmp2}\,(\mathit{i1},\ n,\ \mathit{lst1},\ \mathit{i2},\ \text{mcdr}\,(j,\ \mathit{lst2})) = \text{strncmp2}\,(\mathit{i1},\ n,\ \mathit{lst1},\ \mathit{i2} + j,\ \mathit{lst2})$

```
; a lemma of strlen and mcdr.
```

THEOREM: strlen-cdr
$(i \in \mathbf{N}) \rightarrow (\text{strlen}\,(i,\ n - 1,\ \text{mcdr}\,(\mathtt{1},\ \mathit{lst})) = (\text{strlen}\,(1 + i,\ n,\ \mathit{lst}) - 1))$

THEOREM: strlen-mcdr
$(i \in \mathbf{N})$
$\rightarrow \quad (\text{strlen}\,(i,\ n,\ \text{mcdr}\,(k,\ \mathit{lst})) = (\text{strlen}\,(i + k,\ k + n,\ \mathit{lst}) - k))$

```
; some properties of strstr.
; a lemma for strstr-thm1.
```

THEOREM: strstr1-thm1
let j **be** $\text{strstr1}\,(i,\ \mathit{n1},\ \mathit{lst1},\ \mathit{n2},\ \mathit{lst2},\ \mathit{len})$
in
$((j \in \mathbf{N}) \wedge (n = (1 + \mathit{len})))$
$\rightarrow \quad (\text{strncmp2}\,(j,\ n,\ \mathit{lst1},\ \mathtt{0},\ \mathit{lst2}) = \mathtt{0})$ **endlet**

```
; all 0 <= i < (strlen lst2), lst1[j+i] == lst2[i], if strstr == j =\= 0.
```

THEOREM: strstr-thm1
let j **be** $\text{strstr}\,(\mathit{n1},\ \mathit{lst1},\ \mathit{n2},\ \mathit{lst2})$
in
$(j \in \mathbf{N}) \rightarrow (\text{strncmp}\,(\text{strlen}\,(\mathtt{0},\ \mathit{n2},\ \mathit{lst2}),\ \text{mcdr}\,(j,\ \mathit{lst1}),\ \mathit{lst2}) = \mathtt{0})$ **endlet**

```
; a few lemmas about strchr1.
```

THEOREM: strchr1-nth
$((i \leq j) \wedge (j < \text{strlen}\,(i,\ n,\ \mathit{lst})))$
$\rightarrow \quad (\text{strchr1}\,(i,\ n,\ \mathit{lst},\ \text{get-nth}\,(j,\ \mathit{lst})) \in \mathbf{N})$

THEOREM: strlen-strchr1-nth
$((i \leq j) \land (j < \text{strlen}(i,\ n,\ lst)))$
$\rightarrow \quad (\text{strlen}(1 + \text{strchr1}(i,\ n,\ lst,\ \text{get-nth}(j,\ lst)),\ n,\ lst)$
$\qquad = \quad \text{strlen}(i,\ n,\ lst))$

THEOREM: strlen-strchr1
$(\text{strchr1}(i,\ n,\ lst,\ \mathtt{0}) \in \mathbf{N}) \rightarrow (\text{strchr1}(i,\ n,\ lst,\ \mathtt{0}) \not< \text{strlen}(i,\ n,\ lst))$

THEOREM: strchr1-ch0
$\neg\ \text{strchr1}(i,\ n,\ lst,\ \mathtt{0})$

THEOREM: strchr1-nth-first
$((\text{strchr1}(i,\ n,\ lst,\ \text{get-nth}(j,\ lst)) \in \mathbf{N}) \land (i \leq j))$
$\rightarrow \quad (j \not< \text{strchr1}(i,\ n,\ lst,\ \text{get-nth}(j,\ lst)))$

```
; a lemma for strstr-thm2.
```

THEOREM: strstr1-thm2
$(\text{lst-of-chrp}(lst1)$
$\land \quad \text{lst-of-chrp}(lst2)$
$\land \quad (i \leq j)$
$\land \quad (j < \text{strstr1}(i,\ n1,\ lst1,\ n2,\ lst2,\ len))$
$\land \quad (n = (1 + len)))$
$\rightarrow \quad (\text{strncmp2}(j,\ n,\ lst1,\ \mathtt{0},\ lst2) \neq \mathtt{0})$

```
; all j < (strstr lst1 lst2), (strncmp (mcdr j lst1) lst2) =\= 0.
```

THEOREM: strstr-thm2
$(\text{lst-of-chrp}(lst1)$
$\land \quad \text{lst-of-chrp}(lst2)$
$\land \quad (j < \text{strstr}(n1,\ lst1,\ n2,\ lst2))$
$\land \quad (n2 \not\simeq \mathtt{0}))$
$\rightarrow \quad (\text{strncmp}(\text{strlen}(\mathtt{0},\ n2,\ lst2),\ \text{mcdr}(j,\ lst1),\ lst2) \neq \mathtt{0})$

```
; a lemma for strstr-thm3.
```

THEOREM: strstr1-thm3
$(\text{lst-of-chrp}(lst1)$
$\land \quad \text{lst-of-chrp}(lst2)$
$\land \quad (\neg\ \text{strstr1}(i,\ n1,\ lst1,\ n2,\ lst2,\ len))$
$\land \quad (i \leq j)$
$\land \quad (j < \text{strlen}(i,\ n1,\ lst1))$
$\land \quad (n = (1 + len)))$
$\rightarrow \quad (\text{strncmp2}(j,\ n,\ lst1,\ \mathtt{0},\ lst2) \neq \mathtt{0})$

```
; all j < (strlen lst1), (strncmp (mcdr j lst1) lst2) =\= 0.
```

THEOREM: strstr-thm3
$(\text{lst-of-chrp}\,(lst1)$
$\land\ \ \text{lst-of-chrp}\,(lst2)$
$\land\ \ (\neg\ \text{strstr}\,(n1,\ lst1,\ n2,\ lst2))$
$\land\ \ (j < \text{strlen}\,(\mathtt{0},\ n1,\ lst1))$
$\land\ \ (n2 \not\simeq \mathtt{0}))$
$\to\ \ (\text{strncmp}\,(\text{strlen}\,(\mathtt{0},\ n2,\ lst2),\ \text{mcdr}\,(j,\ lst1),\ lst2) \neq \mathtt{0})$

```
; some properties of strtok.
```

THEOREM: strtok-thm1
let i **be** $\text{strspn}\,(\mathtt{0},\ n1,\ lst1,\ n2,\ lst2)$
in
$((\text{nat-to-uint}\,(s1) \neq \mathtt{0}) \land (\text{get-nth}\,(i,\ lst1) = \mathtt{0}))$
$\to\ \ ((\text{strtok-tok}\,(s1,\ last,\ n1,\ lst1,\ n2,\ lst2) = \mathtt{0})$
$\quad\land\ \ (\text{strtok-last}\,(s1,\ last,\ n1,\ lst1,\ n2,\ lst2) = \mathtt{0})$
$\quad\land\ \ (\text{strtok-lst}\,(n1,\ lst1,\ n2,\ lst2) = lst1))$ **endlet**

THEOREM: strtok-thm2
let i **be** $\text{strspn}\,(\mathtt{0},\ n1,\ lst1,\ n2,\ lst2)$
in
$((\text{nat-to-uint}\,(s1) \neq \mathtt{0})$
$\land\ \ (\text{get-nth}\,(i,\ lst1) \neq \mathtt{0})$
$\land\ \ (\text{get-nth}\,(\text{strcspn}\,(i,\ n1,\ lst1,\ n2,\ lst2),\ lst1) = \mathtt{0})$
$\land\ \ \text{stringp}\,(\mathtt{0},\ n1,\ lst1)$
$\land\ \ (n1 \in \mathbf{N}))$
$\to\ \ ((\text{strtok-tok}\,(s1,\ last,\ n1,\ lst1,\ n2,\ lst2)$
$\quad\ \ =\ \ \text{add}\,(\mathtt{32},\ s1,\ \text{strspn*}\,(\mathtt{0},\ \mathtt{0},\ n1,\ lst1,\ n2,\ lst2)))$
$\quad\land\ \ (\text{strtok-last}\,(s1,\ last,\ n1,\ lst1,\ n2,\ lst2) = \mathtt{0})$
$\quad\land\ \ (\text{strtok-lst}\,(n1,\ lst1,\ n2,\ lst2) = lst1))$ **endlet**

THEOREM: strspn-strchr1
$\text{strspn}\,(i1,\ n1,\ lst1,\ n2,\ lst2)$
$\to\ \ (\neg\ \text{strchr1}\,(\mathtt{0},\ n2,\ lst2,\ \text{get-nth}\,(\text{strspn}\,(i1,\ n1,\ lst1,\ n2,\ lst2),\ lst1)))$

THEOREM: strchr-strchr1
$(ch \neq \mathtt{0}) \to (\text{strchr}\,(i,\ n,\ lst,\ ch) = \text{strchr1}\,(i,\ n,\ lst,\ ch))$

THEOREM: strspn-true
$(\text{stringp}\,(i,\ n1,\ lst1) \land \text{stringp}\,(\mathtt{0},\ n2,\ lst2)) \to \text{strspn}\,(i,\ n1,\ lst1,\ n2,\ lst2)$

THEOREM: strtok-thm3
let i^* **be** $\text{strspn*}\,(\mathtt{0},\ \mathtt{0},\ n1,\ lst1,\ n2,\ lst2)$,
$\quad i$ **be** $\text{strspn}\,(\mathtt{0},\ n1,\ lst1,\ n2,\ lst2)$
in

$$
\begin{array}{l}
((\text{nat-to-uint}\,(s1) \neq \mathtt{0}) \\
\;\wedge\; (\text{get-nth}\,(i,\, lst1) \neq \mathtt{0}) \\
\;\wedge\; (\text{get-nth}\,(\text{strcspn}\,(i,\, n1,\, lst1,\, n2,\, lst2),\, lst1) \neq \mathtt{0}) \\
\;\wedge\; \text{stringp}\,(\mathtt{0},\, n1,\, lst1) \\
\;\wedge\; \text{stringp}\,(\mathtt{0},\, n2,\, lst2) \\
\;\wedge\; (n1 \in \mathbf{N})) \\
\rightarrow\; ((\text{strtok-tok}\,(s1,\, last,\, n1,\, lst1,\, n2,\, lst2) \\
\qquad = \; \text{add}\,(\mathtt{32},\, s1,\, \text{strspn*}\,(\mathtt{0},\, \mathtt{0},\, n1,\, lst1,\, n2,\, lst2))) \\
\qquad \wedge\; (\text{strtok-last}\,(s1,\, last,\, n1,\, lst1,\, n2,\, lst2) \\
\qquad\qquad = \; \text{add}\,(\mathtt{32}, \\
\qquad\qquad\qquad s1, \\
\qquad\qquad\qquad \text{add}\,(\mathtt{32},\, \text{strcspn*}\,(i*,\, i,\, n1,\, lst1,\, n2,\, lst2),\, \mathtt{1}))) \\
\qquad \wedge\; (\text{get-nth}\,(\text{strcspn}\,(i,\, n1,\, lst1,\, n2,\, lst2), \\
\qquad\qquad\qquad \text{strtok-lst}\,(n1,\, lst1,\, n2,\, lst2)) \\
\qquad\qquad = \; \mathtt{0}))\ \textbf{endlet}
\end{array}
$$

THEOREM: strtok-thm4

$$
\begin{array}{l}
(\text{strtok-tok}\,(\mathtt{0},\, \mathtt{0},\, n1,\, lst1,\, n2,\, lst2) = \mathtt{0}) \\
\wedge\; (\text{strtok-last}\,(\mathtt{0},\, \mathtt{0},\, n1,\, lst1,\, n2,\, lst2) = \mathtt{0})
\end{array}
$$

THEOREM: strtok-thm5

$$
\begin{array}{l}
\textbf{let}\ i\ \ \textbf{be}\ \text{strspn}\,(\mathtt{0},\, n1,\, lst1,\, n2,\, lst2) \\
\textbf{in} \\
((\text{nat-to-uint}\,(last) \neq \mathtt{0}) \\
\;\wedge\; (\text{get-nth}\,(i,\, lst1) \neq \mathtt{0}) \\
\;\wedge\; (\text{get-nth}\,(\text{strcspn}\,(i,\, n1,\, lst1,\, n2,\, lst2),\, lst1) = \mathtt{0}) \\
\;\wedge\; \text{stringp}\,(\mathtt{0},\, n1,\, lst1) \\
\;\wedge\; (n1 \in \mathbf{N})) \\
\rightarrow\; ((\text{strtok-tok}\,(\mathtt{0},\, last,\, n1,\, lst1,\, n2,\, lst2) \\
\qquad = \; \text{add}\,(\mathtt{32},\, last,\, \text{strspn*}\,(\mathtt{0},\, \mathtt{0},\, n1,\, lst1,\, n2,\, lst2))) \\
\qquad \wedge\; (\text{strtok-last}\,(\mathtt{0},\, last,\, n1,\, lst1,\, n2,\, lst2) = \mathtt{0}) \\
\qquad \wedge\; (\text{strtok-lst}\,(n1,\, lst1,\, n2,\, lst2) = lst1))\ \textbf{endlet}
\end{array}
$$

THEOREM: strtok-thm6

$$
\begin{array}{l}
\textbf{let}\ i*\ \ \textbf{be}\ \text{strspn*}\,(\mathtt{0},\, \mathtt{0},\, n1,\, lst1,\, n2,\, lst2), \\
\quad\ i\ \ \textbf{be}\ \text{strspn}\,(\mathtt{0},\, n1,\, lst1,\, n2,\, lst2) \\
\textbf{in} \\
((\text{nat-to-uint}\,(last) \neq \mathtt{0}) \\
\;\wedge\; (\text{get-nth}\,(i,\, lst1) \neq \mathtt{0}) \\
\;\wedge\; (\text{get-nth}\,(\text{strcspn}\,(i,\, n1,\, lst1,\, n2,\, lst2),\, lst1) \neq \mathtt{0}) \\
\;\wedge\; \text{stringp}\,(\mathtt{0},\, n1,\, lst1) \\
\;\wedge\; \text{stringp}\,(\mathtt{0},\, n2,\, lst2) \\
\;\wedge\; (n1 \in \mathbf{N})) \\
\rightarrow\; ((\text{strtok-tok}\,(\mathtt{0},\, last,\, n1,\, lst1,\, n2,\, lst2) \\
\qquad = \; \text{add}\,(\mathtt{32},\, last,\, \text{strspn*}\,(\mathtt{0},\, \mathtt{0},\, n1,\, lst1,\, n2,\, lst2)))
\end{array}
$$

$\wedge$ (strtok-last (0, *last*, *n1*, *lst1*, *n2*, *lst2*)
$=$ add (32,
last,
add (32, strcspn* (i^*, i, *n1*, *lst1*, *n2*, *lst2*), 1)))
$\wedge$ (get-nth (strcspn (i, *n1*, *lst1*, *n2*, *lst2*),
strtok-lst (*n1*, *lst1*, *n2*, *lst2*))
$=$ 0)) **endlet**

EVENT: Disable strchr-strchr1.

```
; some properties of memmove.
; memmove is equivalent to lstmov.
```

THEOREM: mmov1-lst-cpy
mmov1-lst (i, *lst1*, *lst2*, n) $=$ lstcpy (i, *lst1*, i, *lst2*, n)

THEOREM: mmovn-mmov1-lst
mmovn-lst (h, *lst1*, *lst2*, i, n) $=$ mmov1-lst (i, *lst1*, *lst2*, $h * n$)

THEOREM: lstcpy-nonnumberp
($i1 \notin \mathbf{N}$)
$\rightarrow$ ((lstcpy ($i1$, *lst1*, $i2$, *lst2*, n) $=$ lstcpy (0, *lst1*, $i2$, *lst2*, n))
$\wedge$ (lstcpy ($i2$, *lst1*, $i1$, *lst2*, n) $=$ lstcpy ($i2$, *lst1*, 0, *lst2*, n)))

THEOREM: lstcpy-lstcpy-0
lstcpy ($h1$, lstcpy (0, *lst1*, 0, *lst2*, $h1$), $h1$, *lst2*, $h2$)
$=$ lstcpy (0, *lst1*, 0, *lst2*, $h1 + h2$)

THEOREM: mmov1-lst1-cpy
($n \leq i$)
$\rightarrow$ (mmov1-lst1 (i, *lst1*, *lst2*, n) $=$ lstcpy ($i - n$, *lst1*, $i - n$, *lst2*, n))

THEOREM: lstcpy-lstcpy-1
(($j1 = (h1 + i1)$) $\wedge$ ($j2 = (h1 + i2)$))
$\rightarrow$ (lstcpy ($i1$, lstcpy ($j1$, *lst1*, $j2$, *lst2*, $h2$), $i2$, *lst2*, $h1$)
$=$ lstcpy ($i1$, *lst1*, $i2$, *lst2*, $h1 + h2$))

THEOREM: times-lessp-dual
$(x < y) \rightarrow ((x < (y * z)) = (z \not\simeq 0))$

THEOREM: mmovn-lst1-mmov1-lst
($(h * n) \leq i$)
$\rightarrow$ (mmovn-lst1 (h, *lst1*, *lst2*, i, n)
$=$ mmov1-lst ($i - (h * n)$, *lst1*, *lst2*, $h * n$))

THEOREM: quotient-shrink-fast-1
$(x < (y * (x \div y))) = \mathbf{f}$

THEOREM: memmove-mmov1-lst
memmove ($str1$, $str2$, n, $lst1$, $lst2$)
$=$ **if** $n \simeq$ `0` **then** $lst1$
elseif nat-to-uint ($str1$) = nat-to-uint ($str2$) **then** $lst2$
else mmov1-lst (`0`, $lst1$, $lst2$, n) **endif**

```
; for k < i, lstcpy(lst1,lst2)[k] == lst1[k].
```

THEOREM: lstcpy-get-1
$(k < i) \rightarrow$ (get-nth (k, lstcpy (i, $lst1$, j, $lst2$, n)) = get-nth (k, $lst1$))

```
; for i <= i1 < i+n, lstcpy(lst1,lst2)[i1] == lst2[i1].
```

THEOREM: lstcpy-get-2
$((i \leq i1) \land (i1 < (i + n)))$
$\rightarrow$ (get-nth ($i1$, lstcpy (i, $lst1$, j, $lst2$, n))
$=$ get-nth ($(j + i1) - i$, $lst2$))

```
; for 0 <= i < n, mmov1-lst(lst1,lst2)[i] == lst2[i].
```

THEOREM: mmov1-lst-thm1
$(i1 < n) \rightarrow$ (get-nth ($i1$, mmov1-lst (`0`, $lst1$, $lst2$, n)) = get-nth ($i1$, $lst2$))

```
; for 0 <= i < n, memmove(lst1,lst2,n)[i] == lst2[i].
```

THEOREM: memmove-thm1
$(i < n)$
$\rightarrow$ (get-nth (i, memmove ($str1$, $str2$, n, $lst1$, $lst2$)) = get-nth (i, $lst2$))

Index