Verifying Memory Managers using GSep

John Wickerson

February 2, 2011

Preliminary: Spatial closure operators

Suppose R and S are of type $loc \to loc \to assertion$. Define:

$$\begin{aligned}
R; S &\stackrel{\text{def}}{=} \lambda x\, z.\, \exists y.\, R\, x\, y \;*\; S\, y\, z \\
R \vee S &\stackrel{\text{def}}{=} \lambda x\, y.\, R\, x\, y \vee S\, x\, y \\
id &\stackrel{\text{def}}{=} \lambda x\, y.\, x = y \wedge emp \\
R^* &\stackrel{\text{def}}{=} \mu S.\, S = id \vee R; S \\
R^+ &\stackrel{\text{def}}{=} R; R^*
\end{aligned}$$

Then the ordinary *list* predicate can be defined like so:

$$list(x) \stackrel{\text{def}}{=} (\lambda x\, y.\, x \mapsto y)^*\, x\, 0$$

Furthermore, we can parameterise the definitions by an element m of a partial commutative monoid (PCM) $(M, \cdot, u)$. Define:

$$\begin{aligned}
R; S &\stackrel{\text{def}}{=} \lambda x\, z\, m.\, \exists y\, m_1\, m_2.\, m = m_1 \cdot m_2 \;*\; R\, x\, y\, m_1 \;*\; S\, y\, z\, m_2 \\
R \vee S &\stackrel{\text{def}}{=} \lambda x\, y\, m.\, R\, x\, y\, m \vee S\, x\, y\, m \\
id &\stackrel{\text{def}}{=} \lambda x\, y\, m.\, x = y \wedge m = u \wedge emp \\
R^* &\stackrel{\text{def}}{=} \mu S.\, S = id \vee R; S \\
R^+ &\stackrel{\text{def}}{=} R; R^*
\end{aligned}$$

Firstly, using the PCM of sets of naturals, $(\mathcal{P}\,\mathbb{N}, \uplus, \emptyset)$, we can define Bornat-style lists, which are parameterised by the set X of locations through which they pass, like so:

$$blist(x, X) \stackrel{\text{def}}{=} (\lambda x\, y\, X.\, x \mapsto y \wedge X = \{x\})^*\, x\, 0\, X$$

Secondly, using the unique 1-element PCM, $(\{u\}, \lambda\_\ \_.\, u, u)$, the extra parameters become redundant, and can be removed in such a way as to restore the original version above.

Thirdly, we can define an arena that comprises a chain of unallocated, allocated, and system blocks – more on this later.

Lemma 1. $R^* = (R^*; R^*)$.

1 A fixed-sized allocator

External spec

$$\vdash \left\{ emp \right\} \texttt{malloc()} \left\{ \texttt{ret} \mapsto \_\ \_ \right\}$$
$$\vdash \left\{ \texttt{x} \mapsto \_\ \_ \right\} \texttt{free(x)} \left\{ emp \right\}$$

1.1 Implementation using a free list

Internal spec

The external spec can be derived from the following internal spec using the hypothetical frame rule.

$$\Delta \vdash \left\{ list\,\texttt{f} \right\} \texttt{malloc()} \left\{ \texttt{ret} \mapsto \_\ \_ * list\,\texttt{f} \right\}$$
$$\Delta \vdash \left\{ \texttt{x} \mapsto \_\ \_ * list\,\texttt{f} \right\} \texttt{free(x)} \left\{ list\,\texttt{f} \right\}$$

where Δ defines:

$$list\,x \stackrel{\text{def}}{=} x = 0 \wedge emp \ \vee\ \exists y.\, x \mapsto y\ \_ * list\,y$$

Verification

```
union list {
  union list* ptr;
  long x;
};

static union list f = NULL;
```
$\left\{ list\,\texttt{f} \right\}$

```
void* malloc()
```
$\left\{ list\,\texttt{f} \right\}$
```
{
```
  $\left\{ \texttt{f} = 0 \wedge emp \ \vee\ \exists y.\, \texttt{f} \mapsto y\ \_ * list\,y \right\}$
```
  void* x;
  if (f == NULL)
```
    $\left\{ \texttt{f} = 0 \wedge emp \right\}$
```
    x = cons(2);
```
    $\left\{ \texttt{x} \mapsto \_\ \_ * (\texttt{f} = 0 \wedge emp) \right\}$

    $\left\{ \texttt{x} \mapsto \_\ \_ * list\,\texttt{f} \right\}$

```
  else {
```
$$\left\{\exists y.\, \mathtt{f} \mapsto y\ \_ * \mathit{list}\, y\right\}$$
```
    x = f;
```
$$\left\{\exists y.\, \mathtt{x} \mapsto y\ \_ * \mathit{list}\, y\right\}$$
```
    f = x->ptr;
```
$$\left\{\mathtt{x} \mapsto \mathtt{f}\ \_ * \mathit{list}\, \mathtt{f}\right\}$$
$$\left\{\mathtt{x} \mapsto \_\ \_ * \mathit{list}\, \mathtt{f}\right\}$$
```
  }
```
$$\left\{\mathtt{x} \mapsto \_\ \_ * \mathit{list}\, \mathtt{f}\right\}$$
```
  return (void *)x;
```
$$\left\{\mathsf{false}\right\}$$
```
}
```
$$\left\{\mathtt{ret} \mapsto \_\ \_ * \mathit{list}\, \mathtt{f}\right\}$$

```
void free(void* ax)
```
$$\left\{\mathtt{ax} \mapsto \_\ \_ * \mathit{list}\, \mathtt{f}\right\}$$
```
{
  List* x = (List*)ax;
```
$$\left\{\mathtt{x} \mapsto \_\ \_ * \mathit{list}\, \mathtt{f}\right\}$$
```
  x -> ptr = f;
```
$$\left\{\mathtt{x} \mapsto \mathtt{f}\ \_ * \mathit{list}\, \mathtt{f}\right\}$$
$$\left\{\exists y.\, \mathtt{x} \mapsto y\ \_ * \mathit{list}\, y\right\}$$
$$\left\{\mathit{list}\, \mathtt{x}\right\}$$
```
  f = x;
}
```
$$\left\{\mathit{list}\, \mathtt{f}\right\}$$

2 A variable-sized allocator

External spec

$$\vdash \left\{\mathit{emp}\right\}\ \mathtt{malloc(n)}\ \left\{(\mathit{token}\, \mathtt{ret}\ \lceil \mathtt{n/WORD} \rceil \ * \ \bigstar_{i=0}^{\lceil \mathtt{n/WORD} \rceil - 1}.\, \mathtt{ret} + i \mapsto \_) \vee \mathtt{ret} = 0\right\}$$
$$\vdash \left\{\exists n.\, \mathit{token}\, \mathtt{x}\, n \ * \ \bigstar_{i=0}^{n-1}.\, \mathtt{x} + i \mapsto \_\right\}\ \mathtt{free(x)}\ \left\{\mathit{emp}\right\}$$

2.1 Naïve implementation (no free list)

Internal spec

The external spec can be derived from the following internal spec using the rule for weakening predicate environments.

$$\Delta \vdash \left\{ emp \right\} \texttt{x := malloc(n)} \left\{ token\, \texttt{ret}\, \texttt{n} \ * \ \circledast_{i=0}^{\texttt{n}-1} .\, \texttt{ret} + i \mapsto \_ \right\}$$
$$\Delta \vdash \left\{ \exists n.\, token\, \texttt{x}\, n \ * \ \circledast_{i=0}^{n-1} .\, \texttt{x} + i \mapsto \_ \right\} \texttt{free(x)} \left\{ emp \right\}$$

where Δ defines:

$$token\, x\, n \ \stackrel{\text{def}}{=} \ (x-1) \mapsto n$$

Verification

```
void* malloc(int n)
```
$\left\{ emp \right\}$
```
{
  void* x = cons(n+1);
```
  $\left\{ \circledast_{i=0}^{\texttt{n}} .\, (\texttt{x} + i) \mapsto \_ \right\}$
```
  *x = n;
```
  $\left\{ \texttt{x} \mapsto \texttt{n} \ * \ \circledast_{i=1}^{\texttt{n}} .\, (\texttt{x} + i) \mapsto \_ \right\}$
  $\left\{ \texttt{x} \mapsto \texttt{n} \ * \ \circledast_{i=0}^{\texttt{n}-1} .\, (\texttt{x} + 1 + i) \mapsto \_ \right\}$
  $\left\{ token\, (\texttt{x} + 1)\, \texttt{n} \ * \ \circledast_{i=0}^{\texttt{n}-1} .\, (\texttt{x} + 1 + i) \mapsto \_ \right\}$
```
  return x+1;
```
  $\left\{ \mathsf{false} \right\}$
```
}
```
$\left\{ token\, \texttt{ret}\, \texttt{n} \ * \ \circledast_{i=0}^{\texttt{n}-1} .\, (ret + i) \mapsto \_ \right\}$

```
void free(void* x)
```
$\left\{ \exists n.\, token\, \texttt{x}\, n \ * \ \circledast_{i=0}^{n-1} .\, (\texttt{x} + i) \mapsto \_ \right\}$
```
{
```
  $\left\{ \exists n.\, (\texttt{x} - 1) \mapsto n \ * \ \circledast_{i=0}^{n-1} .\, (\texttt{x} + i) \mapsto \_ \right\}$
```
  int n = *(x-1);
```
  $\left\{ (\texttt{x} - 1) \mapsto \texttt{n} \ * \ \circledast_{i=0}^{\texttt{n}-1} .\, (\texttt{x} + i) \mapsto \_ \right\}$
  $\left\{ \circledast_{i=-1}^{\texttt{n}-1} .\, (\texttt{x} + i) \mapsto \_ \right\}$

$$\left\{ \circledast_{i=-1}^{\mathtt{n}-1} .\, (\mathtt{x}+i) \mapsto \_ \right\}$$

```
while (n>=0) {
```

$$\left\{ \mathtt{n} \geq 0 \wedge \circledast_{i=-1}^{\mathtt{n}-1} .\, (\mathtt{x}+i) \mapsto \_ \right\}$$

```
n--;
```

$$\left\{ (\mathtt{x}+\mathtt{n}) \mapsto \_ \;*\; \circledast_{i=-1}^{\mathtt{n}-1} .\, (\mathtt{x}+i) \mapsto \_ \right\}$$

```
dispose(x+n);
```

$$\left\{ \circledast_{i=-1}^{\mathtt{n}-1} .\, (\mathtt{x}+i) \mapsto \_ \right\}$$

```
}
```

$$\left\{ \mathtt{n} < 0 \wedge \circledast_{i=-1}^{\mathtt{n}-1} .\, (\mathtt{x}+i) \mapsto \_ \right\}$$

```
}
```

$$\left\{ emp \right\}$$

2.2 Second implementation (Unix V7)

Note that the various 'pure' operators, such as '=' and '>' and 'def(−)', are all given an empty footprint. That is, read $x = 5$ as $x = 5 \wedge emp$.

The external spec can be derived from the following internal spec using the hypothetical frame rule (which removes the invariant *anArena*), the rule for weakening predicate environments (which removes Δ), and the Erase rule (which removes G).

Internal spec

$$\delta;\gamma;G \vdash \left\{ anArena \right\} \mathtt{malloc(n)} \left\{ \begin{array}{l} anArena \;* \\ ((token\,\mathtt{ret}\,\lceil \mathtt{n}/\mathtt{WORD} \rceil \;*\; \circledast_{i=0}^{\lceil \mathtt{n}/\mathtt{WORD} \rceil -1} .\, \mathtt{ret}+i \mapsto \_) \vee \mathtt{ret}=0) \end{array} \right\}$$

$$\delta;\gamma;G \vdash \left\{ anArena \;*\; \exists n.\, token\,\mathtt{x}\,n \;*\; \circledast_{i=0}^{n-1} .\, \mathtt{x}+i \mapsto \_ \right\} \mathtt{free(x)} \left\{ anArena \right\}$$

where δ defines:

$$\begin{array}{rcl}
ublock\,x\,y\,B & \stackrel{\text{def}}{=} & B = \{x+1 \mapsto_{\mathsf{u}} y-x-1\} \;*\; x<y \;*\; x \mapsto y \;*\; \circledast_{i=x+1}^{y-1} .\, i \mapsto _ \\
ablock\,x\,y\,B & \stackrel{\text{def}}{=} & B = \{x+1 \mapsto_{\mathsf{a}} y-x-1\} \;*\; x<y \;*\; x_{|1} \stackrel{.5}{\mapsto} y \\
sblock\,x\,y\,B & \stackrel{\text{def}}{=} & B = \{x+1 \mapsto_{\mathsf{s}} y-x-1\} \;*\; x<y \;*\; x_{|1} \mapsto y \\
block & \stackrel{\text{def}}{=} & ublock \vee ablock \vee sblock \\
uninit\;A & \stackrel{\text{def}}{=} & \mathtt{s} \mapsto 0\,0 \;*\; A=\emptyset \;*\; brka(\mathtt{s}+2) \\
arena\;A & \stackrel{\text{def}}{=} & \exists B_1, B_2 : \mathcal{B}.\, block^{*}\;\mathtt{s}\,\mathtt{v}\,B_1 \;*\; block^{*}\;\mathtt{v}\,\mathtt{t}\,B_2 \\
 & & *\; A = (B_1 \uplus B_2)^{\mathsf{a}} \;*\; \mathtt{t}_{|1} \mapsto \mathtt{s} \;*\; brka(\mathtt{t}+1) \\
anArena & \stackrel{\text{def}}{=} & \boxed{\exists A.\, uninit\;A \;\vee\; arena\;A} \\
token\,x\,n & \stackrel{\text{def}}{=} & \boxed{\exists A.\, arena(A \uplus \{x \mapsto n\})} \;*\; (x-1)_{|1} \stackrel{.5}{\mapsto} x+n
\end{array}$$

Note that we use the following separation algebra for the spatial closure operators:

$$\mathcal{B} \stackrel{\text{def}}{=} (\mathbb{N} \rightharpoonup \{\mathsf{u}, \mathsf{a}, \mathsf{s}\} \times \mathbb{N}_0, \uplus, \emptyset)$$

Note also that B^{a} returns a function of type $\mathbb{N} \rightharpoonup \mathbb{N}_0$, such that $(x \mapsto n) \in B^{\mathsf{a}}$ if and only if $(x \mapsto_{\mathsf{a}} n) \in B$.

The guarantee G is defined as $\bigcup_x \{\mathit{Malloc}, \mathit{Free}\, x\}$, where:

$$\begin{aligned} \mathit{Malloc} &\stackrel{\text{def}}{=} \exists A, x, n.\, (\mathsf{s} \mapsto 0\, 0 \ * \ A = \emptyset) \ \vee \ \mathit{arena}\, A \ \rightsquigarrow \ \mathit{arena}(A \uplus \{x \mapsto n\}) \\ \mathit{Free}\, x &\stackrel{\text{def}}{=} \exists A, n.\, (x-1)_{|1} \stackrel{.5}{\mapsto} (x+n) \ \mid \ \mathit{arena}(A \uplus \{x \mapsto n\}) \ \rightsquigarrow \ \mathit{arena}\, A \end{aligned}$$

The procedure environment γ provides a specification for `sbrk`. The 'official' spec for `sbrk`is as follows:

$$\vdash \left\{ \mathit{brk}(b) \right\} \mathtt{sbrk(n)} \left\{ \begin{array}{l} (\mathit{brk}(b) \ * \ \mathtt{ret} = -1 \ * \ \mathtt{n} \neq 0) \vee \\ (\mathit{brk}(b + \lceil \mathtt{n}/\mathtt{WORD} \rceil) \ * \ \mathtt{ret} = b \ * \ \bigast_{i=0}^{\lceil \mathtt{n}/\mathtt{WORD} \rceil - 1} .\, \mathtt{ret} + i \mapsto \_) \end{array} \right\}$$

but if we define $\mathit{brka}(x)$ as shorthand for $\exists b \geq x.\, \mathit{brk}(b)$, then we obtain the following derived spec:

$$\vdash \left\{ \mathit{brka}(x) \right\} \mathtt{sbrk(n)} \left\{ \begin{array}{l} (\mathit{brka}(x) \ * \ \mathtt{ret} = -1 \ * \ \mathtt{n} \neq 0) \vee \\ (\mathit{brka}(\mathtt{ret} + \lceil \mathtt{n}/\mathtt{WORD} \rceil) \ * \ x \leq \mathtt{ret} \ * \ \bigast_{i=0}^{\lceil \mathtt{n}/\mathtt{WORD} \rceil - 1} .\, \mathtt{ret} + i \mapsto \_) \end{array} \right\}$$

which is easier to use, and is hence the one contained in γ.

The verification of the module depends on the following two lemmas:

Lemma 2. $\mathit{block}^*\, x_1\, y_1\, B_1 \ * \ \mathit{block}^*\, x_2\, y_2\, B_2 \implies B_1 \perp B_2$

Lemma 3. $\mathit{block}^*\, x\, y\, B \ * \ w \mapsto z \implies w + 1 \notin \mathrm{dom}(B)$

Verification of malloc routine

```
#define WORD sizeof(union store)
#define BLOCK 1024 /* a multiple of WORD*/
#define testbusy(p) ((int)(p)&1)
#define setbusy(p) (struct store *)((int)(p)|1)
#define clearbusy(p) (struct store *)((int)(p)&~1)

struct store {struct store *ptr;};
static struct store s[2]; /* initial arena */
static struct store *v; /* search ptr */
static struct store *t; /* arena top */

char *malloc(unsigned int nbytes)
```

$$\left\{ \mathit{anArena} \right\}$$

$$\left\{ \boxed{\exists A.\, \mathit{uninit}\, A \ \vee \ \mathit{arena}\, A} \right\}$$

```
// begin Existential
```

$$\left\{ \boxed{\mathit{uninit}\, A \ \vee \ \mathit{arena}\, A} \right\}$$

```
// begin region update (action is either Malloc or none)
```

$$\left\{ \mathit{uninit}\, A \ \vee \ \mathit{arena}\, A \right\}$$

```
// Precondition for returning:
```

$$\left\{ \begin{pmatrix} \mathit{arena}(A \uplus \{\mathtt{ret} \mapsto \lceil \mathtt{nbytes/WORD} \rceil\}) \\ * \ \mathop{\bigast}_{i=0}^{\lceil \mathtt{nbytes/WORD} \rceil - 1} . \, \mathtt{ret} + i \mapsto \_ \\ * \ (\mathtt{ret} - 1)_{|1} \overset{.5}{\mapsto} \mathtt{ret} + \lceil \mathtt{nbytes/WORD} \rceil \end{pmatrix} \vee (\mathit{arena}\ A \ * \ \mathtt{ret} = 0) \right\}$$

```
{
```

$$\left\{ \mathit{uninit}\, A \ \vee \ \mathit{arena}\, A \right\}$$

```
  register struct store *p, *q;
  register nw;
  static temp;
  if(s[0].ptr == 0) { /*first time*/
```

$$\left\{ \mathit{uninit}\, A \right\}$$

$$\left\{ \mathtt{s} \mapsto 0\,0 \ * \ \mathit{brka}(\mathtt{s}+2) \ * \ A = \emptyset \right\}$$

```
    s[0].ptr = setbusy(&s[1]);
```

$$\left\{ \mathtt{s}_{|1} \mapsto \mathtt{s}+1 \ * \ \mathtt{s}+1 \mapsto 0 \ * \ \mathit{brka}(\mathtt{s}+2) \ * \ A = \emptyset \right\}$$

```
    s[1].ptr = setbusy(&s[0]);
```

$$\left\{ \mathtt{s}_{|1} \mapsto \mathtt{s}+1 \ * \ (\mathtt{s}+1)_{|1} \mapsto \mathtt{s} \ * \ \mathit{brka}(\mathtt{s}+2) \ * \ A = \emptyset \right\}$$

```
    t = &s[1];
```

$$\left\{ \mathtt{s}_{|1} \mapsto \mathtt{t} \ * \ \mathtt{t}_{|1} \mapsto \mathtt{s} \ * \ \mathtt{s} < \mathtt{t} \ * \ \mathit{brka}(\mathtt{t}+1) \ * \ A = \emptyset \right\}$$

```
    v = &s[0];
```

$$\left\{ \mathtt{s}_{|1} \mapsto \mathtt{t} \ * \ \mathtt{t}_{|1} \mapsto \mathtt{s} \ * \ \mathtt{s} < \mathtt{t} \ * \ \mathtt{v} = \mathtt{s} \ * \ \mathit{brka}(\mathtt{t}+1) \ * \ A = \emptyset \right\}$$

$$\left\{ \mathit{sblock}\, \mathtt{s}\, \mathtt{t}\, \{\mathtt{s}+1 \mapsto_{\mathsf{s}} 0\} \ * \ \mathtt{t}_{|1} \mapsto \mathtt{s} \ * \ \mathtt{v} = \mathtt{s} \ * \ \mathit{brka}(\mathtt{t}+1) \ * \ A = \emptyset \right\}$$

$$\left\{ \begin{array}{l} \exists B_1, B_2.\, \mathit{block}^*\, \mathtt{s}\, \mathtt{v}\, B_1 \ * \ \mathit{block}^*\, \mathtt{v}\, \mathtt{t}\, B_2 \ * \ \mathtt{t}_{|1} \mapsto \mathtt{s} \\ \ * \ A = (B_1 \uplus B_2)^{\mathsf{a}} \ * \ \mathit{brka}(\mathtt{t}+1) \ * \ A = \emptyset \end{array} \right\}$$

$$\left\{ \mathit{arena}\, A \ * \ A = \emptyset \right\}$$

$$\left\{ \mathit{arena}\, A \right\}$$

```
  }
```

$$\left\{\begin{array}{l}\exists B_1, B_2.\, \mathit{block}^{*}\,\mathtt{s}\,\mathtt{v}\,B_1 \;*\; \mathit{block}^{*}\,\mathtt{v}\,\mathtt{t}\,B_2 \;*\; \mathtt{t}_{|1} \mapsto \mathtt{s}\\ *\; A = (B_1 \uplus B_2)^{\mathsf{a}} \;*\; \mathit{brka}(\mathtt{t}+1)\end{array}\right\}$$

```
nw=(nbytes+WORD+WORD-1)/WORD;
```

$$\left\{\begin{array}{l}\exists B_1, B_2.\, \mathit{block}^{*}\,\mathtt{s}\,\mathtt{v}\,B_1 \;*\; \mathit{block}^{*}\,\mathtt{v}\,\mathtt{t}\,B_2 \;*\; \mathtt{t}_{|1} \mapsto \mathtt{s}\\ *\; A = (B_1 \uplus B_2)^{\mathsf{a}} \;*\; \mathit{brka}(\mathtt{t}+1) \;*\; \mathtt{nw} = 1 + \left\lceil \frac{\mathtt{nbytes}}{\mathtt{WORD}} \right\rceil\end{array}\right\}$$

```
for(p=v; ; ) {
  // Loop inv 1:
```

$$\left\{\begin{array}{l}\exists B_1, B_2.\, \mathit{block}^{*}\,\mathtt{s}\,\mathtt{p}\,B_1 \;*\; \mathit{block}^{*}\,\mathtt{p}\,\mathtt{t}\,B_2 \;*\; \mathtt{t}_{|1} \mapsto \mathtt{s}\\ *\; A = (B_1 \uplus B_2)^{\mathsf{a}} \;*\; \mathit{brka}(\mathtt{t}+1) \;*\; \mathtt{nw} = 1 + \left\lceil \frac{\mathtt{nbytes}}{\mathtt{WORD}} \right\rceil\end{array}\right\}$$

```
  for(temp=0; ; ) {
    // Loop inv 2:
```

$$\left\{\begin{array}{l}\exists B_1, B_2.\, \mathit{block}^{*}\,\mathtt{s}\,\mathtt{p}\,B_1 \;*\; \mathit{block}^{*}\,\mathtt{p}\,\mathtt{t}\,B_2 \;*\; \mathtt{t}_{|1} \mapsto \mathtt{s}\\ *\; A = (B_1 \uplus B_2)^{\mathsf{a}} \;*\; \mathit{brka}(\mathtt{t}+1) \;*\; \mathtt{nw} = 1 + \left\lceil \frac{\mathtt{nbytes}}{\mathtt{WORD}} \right\rceil\end{array}\right\}$$

```
    if(!testbusy(p->ptr)) {
```

$$\left\{\begin{array}{l}\exists B_1, B_2, q.\, \mathit{block}^{*}\,\mathtt{s}\,\mathtt{p}\,B_1 \;*\; \mathit{ublock}\,\mathtt{p}\,q\,\{\mathtt{p}+1 \mapsto_{\mathsf{u}} q-\mathtt{p}-1\} \;*\; \mathit{block}^{*}\,q\,\mathtt{t}\,B_2\\ *\; \mathtt{t}_{|1} \mapsto \mathtt{s} \;*\; A = (B_1 \uplus B_2)^{\mathsf{a}} \;*\; \mathit{brka}(\mathtt{t}+1) \;*\; \mathtt{nw} = 1 + \left\lceil \frac{\mathtt{nbytes}}{\mathtt{WORD}} \right\rceil\end{array}\right\}$$

```
      while(!testbusy((q=p->ptr)->ptr)) {
```

$$\left\{\begin{array}{l}\exists B_1, B_2, r.\, \mathit{block}^{*}\,\mathtt{s}\,\mathtt{p}\,B_1 \;*\; \mathit{ublock}\,\mathtt{p}\,\mathtt{q}\,\{\mathtt{p}+1 \mapsto_{\mathsf{u}} \mathtt{q}-\mathtt{p}-1\}\\ *\; \mathit{ublock}\,\mathtt{q}\,r\,\{\mathtt{q}+1 \mapsto_{\mathsf{u}} r-\mathtt{q}-1\} \;*\; \mathit{block}^{*}\,r\,\mathtt{t}\,B_2 \;*\; \mathtt{t}_{|1} \mapsto \mathtt{s}\\ *\; A = (B_1 \uplus B_2)^{\mathsf{a}} \;*\; \mathit{brka}(\mathtt{t}+1) \;*\; \mathtt{nw} = 1 + \left\lceil \frac{\mathtt{nbytes}}{\mathtt{WORD}} \right\rceil\end{array}\right\}$$

```
        p->ptr = q->ptr; // coalesce consecutive free blocks
```

$$\left\{\begin{array}{l}\exists B_1, B_2, r.\, \mathit{block}^{*}\,\mathtt{s}\,\mathtt{p}\,B_1 \;*\; \mathit{ublock}\,\mathtt{p}\,r\,\{\mathtt{p}+1 \mapsto_{\mathsf{u}} r-\mathtt{p}-1\} \;*\; \mathit{block}^{*}\,r\,\mathtt{t}\,B_2\\ *\; \mathtt{t}_{|1} \mapsto \mathtt{s} \;*\; A = (B_1 \uplus B_2)^{\mathsf{a}} \;*\; \mathit{brka}(\mathtt{t}+1) \;*\; \mathtt{nw} = 1 + \left\lceil \frac{\mathtt{nbytes}}{\mathtt{WORD}} \right\rceil\end{array}\right\}$$

```
      }
```

$$\left\{\begin{array}{l}\exists B_1, B_2.\, \mathit{block}^{*}\,\mathtt{s}\,\mathtt{p}\,B_1 \;*\; \mathit{ublock}\,\mathtt{p}\,\mathtt{q}\,\{\mathtt{p}+1 \mapsto_{\mathsf{u}} \mathtt{q}-\mathtt{p}-1\} \;*\; \mathit{block}^{*}\,\mathtt{q}\,\mathtt{t}\,B_2\\ *\; \mathtt{t}_{|1} \mapsto \mathtt{s} \;*\; A = (B_1 \uplus B_2)^{\mathsf{a}} \;*\; \mathit{brka}(\mathtt{t}+1) \;*\; \mathtt{nw} = 1 + \left\lceil \frac{\mathtt{nbytes}}{\mathtt{WORD}} \right\rceil\end{array}\right\}$$

```
      if(q>=p+nw && p+nw>=p) {
```

$$\left\{\begin{array}{l}\exists B_1, B_2.\, \mathit{block}^{*}\,\mathtt{s}\,\mathtt{p}\,B_1 \;*\; \mathit{ublock}\,\mathtt{p}\,\mathtt{q}\,\{\mathtt{p}+1 \mapsto_{\mathsf{u}} \mathtt{q}-\mathtt{p}-1\}\\ *\; \mathit{block}^{*}\,\mathtt{q}\,\mathtt{t}\,B_2 \;*\; \mathtt{t}_{|1} \mapsto \mathtt{s} \;*\; A = (B_1 \uplus B_2)^{\mathsf{a}} \;*\; \mathit{brka}(\mathtt{t}+1)\\ *\; \mathtt{nw} = 1 + \left\lceil \frac{\mathtt{nbytes}}{\mathtt{WORD}} \right\rceil \;*\; \mathtt{q} \geq \mathtt{p} + \mathtt{nw}\end{array}\right\}$$

```
        goto found;
```

$$\left\{\mathsf{false}\right\}$$

```
      }
    }
    // p's block is unavailable / too small,
    // or p points to the top of the arena
```

$$\left\{\begin{array}{l}\exists B_1, B_2.\, \mathit{block}^*\, \mathtt{s}\,\mathtt{p}\, B_1 \ * \ \mathit{block}^*\, \mathtt{p}\,\mathtt{t}\, B_2 \ * \ A = (B_1 \uplus B_2)^{\mathsf{a}} \\ * \ \mathtt{t}_{|1} \mapsto \mathtt{s} \ * \ \mathit{brka}(\mathtt{t}+1) \ * \ \mathtt{nw} = 1 + \left\lceil \frac{\mathtt{nbytes}}{\mathtt{WORD}} \right\rceil \end{array}\right\}$$

```
q = p;
```

$$\left\{\begin{array}{l}\exists B_1, B_2.\, \mathit{block}^*\, \mathtt{s}\,\mathtt{q}\, B_1 \ * \ \mathit{block}^*\, \mathtt{q}\,\mathtt{t}\, B_2 \ * \ A = (B_1 \uplus B_2)^{\mathsf{a}} \\ * \ \mathtt{t}_{|1} \mapsto \mathtt{s} \ * \ \mathit{brka}(\mathtt{t}+1) \ * \ \mathtt{nw} = 1 + \left\lceil \frac{\mathtt{nbytes}}{\mathtt{WORD}} \right\rceil \ * \ \mathtt{q} = \mathtt{p} \end{array}\right\}$$

$$\left\{\begin{array}{l}\exists B_1, B_2.\, \mathit{block}^*\, \mathtt{s}\,\mathtt{q}\, B_1 \ * \ \mathit{block}^*\, \mathtt{q}\,\mathtt{t}\, B_2 \ * \ A = (B_1 \uplus B_2)^{\mathsf{a}} \\ * \ \mathtt{t}_{|1} \mapsto \mathtt{s} \ * \ \mathit{brka}(\mathtt{t}+1) \ * \ \mathtt{nw} = 1 + \left\lceil \frac{\mathtt{nbytes}}{\mathtt{WORD}} \right\rceil \ * \ \mathtt{q} = \mathtt{p} \end{array}\right\}$$

```
p = clearbusy(p->ptr);
```

$$\left\{\begin{array}{l}((\exists B_1, B_2, \tau.\, \mathit{block}^*\, \mathtt{s}\,\mathtt{q}\, B_1 \ * \ \mathit{block}\, \mathtt{q}\,\mathtt{p}\, \{\mathtt{q}+1 \mapsto_\tau \mathtt{p}-\mathtt{q}-1\} \\ * \ \mathit{block}^*\, \mathtt{p}\,\mathtt{t}\, B_2 \ * \ A = (B_1 \uplus \{\mathtt{q}+1 \mapsto_\tau \mathtt{p}-\mathtt{q}-1\} \uplus B_2)^{\mathsf{a}}) \\ \vee (\exists B.\, \mathit{block}^*\, \mathtt{s}\,\mathtt{q}\, B \ * \ A = B^{\mathsf{a}} \ * \ \mathtt{q} = \mathtt{t} \ * \ \mathtt{p} = \mathtt{s})) \\ * \ \mathtt{t}_{|1} \mapsto \mathtt{s} \ * \ \mathit{brka}(\mathtt{t}+1) \ * \ \mathtt{nw} = 1 + \left\lceil \frac{\mathtt{nbytes}}{\mathtt{WORD}} \right\rceil \end{array}\right\}$$

```
if(p>q) {
```

$$\left\{\begin{array}{l}\exists B_1, B_2, \tau.\, \mathit{block}^*\, \mathtt{s}\,\mathtt{q}\, B_1 \ * \ \mathit{block}\, \mathtt{q}\,\mathtt{p}\, \{\mathtt{q}+1 \mapsto_\tau \mathtt{p}-\mathtt{q}-1\} \\ * \ \mathit{block}^*\, \mathtt{p}\,\mathtt{t}\, B_2 \ * \ A = (B_1 \uplus \{\mathtt{q}+1 \mapsto_\tau \mathtt{p}-\mathtt{q}-1\} \uplus B_2)^{\mathsf{a}} \\ * \ \mathtt{t}_{|1} \mapsto \mathtt{s} \ * \ \mathit{brka}(\mathtt{t}+1) \ * \ \mathtt{nw} = 1 + \left\lceil \frac{\mathtt{nbytes}}{\mathtt{WORD}} \right\rceil \end{array}\right\}$$

```
} else if(q!=t || p!=s) {
```

$$\left\{\begin{array}{l}\exists B.\, \mathit{block}^*\, \mathtt{s}\,\mathtt{q}\, B \ * \ \mathtt{t}_{|1} \mapsto \mathtt{s} \ * \ A = B^{\mathsf{a}} \ * \ \mathit{brka}(\mathtt{t}+1) \\ * \ \mathtt{nw} = 1 + \left\lceil \frac{\mathtt{nbytes}}{\mathtt{WORD}} \right\rceil \ * \ \mathtt{q} = \mathtt{t} \ * \ \mathtt{p} = \mathtt{s} \ * \ (\mathtt{q} \neq \mathtt{t} \vee \mathtt{p} \neq \mathtt{s}) \end{array}\right\}$$

$$\{\mathsf{false}\}$$

```
  return 0; // unreachable
```

$$\{\mathsf{false}\}$$

```
} else if(++temp>1) {
```

$$\left\{\begin{array}{l}\exists B.\, \mathit{block}^*\, \mathtt{s}\,\mathtt{q}\, B \ * \ \mathtt{t}_{|1} \mapsto \mathtt{s} \ * \ A = B^{\mathsf{a}} \ * \ \mathit{brka}(\mathtt{t}+1) \\ * \ \mathtt{nw} = 1 + \left\lceil \frac{\mathtt{nbytes}}{\mathtt{WORD}} \right\rceil \ * \ \mathtt{q} = \mathtt{t} \ * \ \mathtt{p} = \mathtt{s} \end{array}\right\}$$

```
  break; // jump to [Extend arena]
```

$$\{\mathsf{false}\}$$

```
}
// Reestablish loop inv 2:
```

$$\left\{\begin{array}{l}\exists B_1, B_2.\, \mathit{block}^*\, \mathtt{s}\,\mathtt{p}\, B_1 \ * \ \mathit{block}^*\, \mathtt{p}\,\mathtt{t}\, B_2 \ * \ A = (B_1 \uplus B_2)^{\mathsf{a}} \\ * \ \mathtt{t}_{|1} \mapsto \mathtt{s} \ * \ \mathit{brka}(\mathtt{t}+1) \ * \ \mathtt{nw} = 1 + \left\lceil \frac{\mathtt{nbytes}}{\mathtt{WORD}} \right\rceil \end{array}\right\}$$

```
}
// We never exit the loop 'normally' (because the non-existent
// test condition never fails). We only reach this point by
// breaking.
// [Extend arena]:
```

$$\left\{\begin{array}{l}\exists B.\, \mathit{block}^{*}\, \mathtt{s}\, \mathtt{t}\, B \ * \ \mathtt{t}_{|1} \mapsto \mathtt{s} \ * \ A = B^{\mathsf{a}} \\ * \ \mathit{brka}(\mathtt{t}+1) \ * \ \mathtt{nw} = 1 + \left\lceil \frac{\mathtt{nbytes}}{\mathtt{WORD}} \right\rceil \ * \ \mathtt{p} = \mathtt{s}\end{array}\right\}$$

```
temp = ((nw+BLOCK/WORD)/(BLOCK/WORD))*(BLOCK/WORD);
```

$$\left\{\begin{array}{l}\exists B.\, \mathit{block}^{*}\, \mathtt{s}\, \mathtt{t}\, B \ * \ \mathtt{t}_{|1} \mapsto \mathtt{s} \ * \ A = B^{\mathsf{a}} \ * \ \mathit{brka}(\mathtt{t}+1) \\ * \ \mathtt{nw} = 1 + \left\lceil \frac{\mathtt{nbytes}}{\mathtt{WORD}} \right\rceil \ * \ \mathtt{p} = \mathtt{s} \ * \ \mathtt{temp} > \mathtt{nw}\end{array}\right\}$$

```
q = (struct store *)sbrk(0);
```

// note that $\mathit{brka}(\mathtt{q}) \implies \mathit{brka}(\mathtt{t}+1)$ because $\mathtt{q} \geq \mathtt{t}+1$

$$\left\{\begin{array}{l}\exists B.\, \mathit{block}^{*}\, \mathtt{s}\, \mathtt{t}\, B \ * \ \mathtt{t}_{|1} \mapsto \mathtt{s} \ * \ A = B^{\mathsf{a}} \ * \ \mathit{brka}(\mathtt{t}+1) \\ * \ \mathtt{nw} = 1 + \left\lceil \frac{\mathtt{nbytes}}{\mathtt{WORD}} \right\rceil \ * \ \mathtt{p} = \mathtt{s} \ * \ \mathtt{temp} > \mathtt{nw} \ * \ \mathtt{q} \geq \mathtt{t}+1\end{array}\right\}$$

```
if(q + temp < q) {
```

$\{\mathsf{false}\}$ // integer overflows aren't modelled

```
  return 0;
```

$\{\mathsf{false}\}$

```
}
```

$$\left\{\begin{array}{l}\exists B.\, \mathit{block}^{*}\, \mathtt{s}\, \mathtt{t}\, B \ * \ \mathtt{t}_{|1} \mapsto \mathtt{s} \ * \ A = B^{\mathsf{a}} \ * \ \mathit{brka}(\mathtt{t}+1) \\ * \ \mathtt{nw} = 1 + \left\lceil \frac{\mathtt{nbytes}}{\mathtt{WORD}} \right\rceil \ * \ \mathtt{p} = \mathtt{s} \ * \ \mathtt{temp} > \mathtt{nw} \ * \ \mathtt{q} \geq \mathtt{t}+1\end{array}\right\}$$

```
q = (struct store *)sbrk(temp * WORD);
```

$$\left\{\begin{array}{l}\exists B.\, \mathit{block}^{*}\, \mathtt{s}\, \mathtt{t}\, B \ * \ \mathtt{t}_{|1} \mapsto \mathtt{s} \ * \ A = B^{\mathsf{a}} \ * \ \mathtt{nw} = 1 + \left\lceil \frac{\mathtt{nbytes}}{\mathtt{WORD}} \right\rceil \\ * \ \mathtt{p} = \mathtt{s} \ * \ \mathtt{temp} > \mathtt{nw} \ * \ ((\mathit{brka}(\mathtt{t}+1) \ * \ \mathtt{q} = -1) \\ \vee \, (\mathit{brka}(\mathtt{q}+\mathtt{temp}) \ * \ \mathtt{t}+1 \leq \mathtt{q} \ * \ \bigstar_{i=0}^{\mathtt{temp}-1}.\, \mathtt{q}+i \mapsto \_))\end{array}\right\}$$

```
if((INT)q == -1) {
```

$\left\{\exists B.\, \mathit{block}^{*}\, \mathtt{s}\, \mathtt{t}\, B \ * \ \mathtt{t}_{|1} \mapsto \mathtt{s} \ * \ A = B^{\mathsf{a}} \ * \ \mathit{brka}(\mathtt{t}+1)\right\}$

```
  v = s; // line added to fix bug
```

$$\left\{\begin{array}{l}\exists B_1, B_2.\, \mathit{block}^{*}\, \mathtt{s}\, \mathtt{v}\, B_1 \ * \ \mathit{block}^{*}\, \mathtt{v}\, \mathtt{t}\, B_2 \ * \ \mathtt{t}_{|1} \mapsto \mathtt{s} \\ * \ A = (B_1 \uplus B_2)^{\mathsf{a}} \ * \ \mathit{brka}(\mathtt{t}+1)\end{array}\right\}$$

$\{\mathit{arena}\, A\}$

$\{(\mathit{arena}\, A \ * \ \mathtt{ret} = 0)[0/\mathtt{ret}]\}$

```
  return 0;
```

$\{\mathsf{false}\}$

```
}
```

$$\left\{\begin{array}{l}\exists B.\, \mathit{block}^{*}\, \mathtt{s}\, \mathtt{t}\, B \ * \ \mathtt{t}_{|1} \mapsto \mathtt{s} \ * \ A = B^{\mathsf{a}} \ * \ \mathtt{nw} = 1 + \left\lceil \frac{\mathtt{nbytes}}{\mathtt{WORD}} \right\rceil \ * \ \mathtt{p} = \mathtt{s} \\ * \ \mathtt{temp} > \mathtt{nw} \ * \ \mathit{brka}(\mathtt{q}+\mathtt{temp}) \ * \ \mathtt{t}+1 \leq \mathtt{q} \ * \ \bigstar_{i=0}^{\mathtt{temp}-1}.\, \mathtt{q}+i \mapsto \_\end{array}\right\}$$

```
t->ptr = q;
```

$$\left\{\begin{array}{l}\exists B.\, \mathit{block}^*\, \mathtt{s}\,\mathtt{t}\, B \ast \mathtt{t} \mapsto \mathtt{q} \ast A = B^{\mathsf{a}} \ast \mathtt{nw} = 1 + \left\lceil \frac{\mathtt{nbytes}}{\mathtt{WORD}} \right\rceil \ast \mathtt{p} = \mathtt{s} \\ \ast\ \mathtt{temp} > \mathtt{nw} \ast \mathit{brka}(\mathtt{q}+\mathtt{temp}) \ast \mathtt{t}+1 \le \mathtt{q} \ast \mathop{\Large\ast}_{i=0}^{\mathtt{temp}-1} .\, \mathtt{q}+i \mapsto \_ \end{array}\right\}$$

```
if(q!=t+1) {
```

$$\left\{\begin{array}{l}\exists B.\, \mathit{block}^*\, \mathtt{s}\,\mathtt{t}\, B \ast \mathtt{t} \mapsto \mathtt{q} \ast A = B^{\mathsf{a}} \ast \mathtt{nw} = 1 + \left\lceil \frac{\mathtt{nbytes}}{\mathtt{WORD}} \right\rceil \ast \mathtt{p} = \mathtt{s} \\ \ast\ \mathtt{temp} > \mathtt{nw} \ast \mathit{brka}(\mathtt{q}+\mathtt{temp}) \ast \mathtt{t}+1 < \mathtt{q} \ast \mathop{\Large\ast}_{i=0}^{\mathtt{temp}-1} .\, \mathtt{q}+i \mapsto \_ \end{array}\right\}$$

```
  t->ptr = setbusy(t->ptr);
```

$$\left\{\begin{array}{l}\exists B.\, \mathit{block}^*\, \mathtt{s}\,\mathtt{t}\, B \ast \mathtt{t}_{|1} \mapsto \mathtt{q} \ast A = B^{\mathsf{a}} \ast \mathtt{nw} = 1 + \left\lceil \frac{\mathtt{nbytes}}{\mathtt{WORD}} \right\rceil \ast \mathtt{p} = \mathtt{s} \\ \ast\ \mathtt{temp} > \mathtt{nw} \ast \mathit{brka}(\mathtt{q}+\mathtt{temp}) \ast \mathtt{t}+1 < \mathtt{q} \ast \mathop{\Large\ast}_{i=0}^{\mathtt{temp}-1} .\, \mathtt{q}+i \mapsto \_ \end{array}\right\}$$

$$\left\{\begin{array}{l}\exists B.\, \mathit{block}^*\, \mathtt{s}\,\mathtt{t}\, B \ast \mathit{sblock}\,\mathtt{t}\,\mathtt{q}\,\{\mathtt{t}+1 \mapsto_{\mathsf{s}} \mathtt{q}-\mathtt{t}-1\} \ast A = B^{\mathsf{a}} \ast \mathtt{nw} = 1 + \left\lceil \frac{\mathtt{nbytes}}{\mathtt{WORD}} \right\rceil \\ \ast\ \mathtt{p} = \mathtt{s} \ast \mathtt{temp} > \mathtt{nw} \ast \mathit{brka}(\mathtt{q}+\mathtt{temp}) \ast \mathop{\Large\ast}_{i=0}^{\mathtt{temp}-1} .\, \mathtt{q}+i \mapsto \_ \end{array}\right\}$$

```
}
// t is either a ublock of size 0 or an sblock
```

$$\left\{\begin{array}{l}\exists B, \tau.\, \mathit{block}^*\, \mathtt{s}\,\mathtt{t}\, B \ast \mathit{block}\,\mathtt{t}\,\mathtt{q}\,\{\mathtt{t}+1 \mapsto_{\tau} \mathtt{q}-\mathtt{t}-1\} \ast A = B^{\mathsf{a}} \ast \mathtt{nw} = 1 + \left\lceil \frac{\mathtt{nbytes}}{\mathtt{WORD}} \right\rceil \\ \ast\ \mathtt{p} = \mathtt{s} \ast \mathtt{temp} > \mathtt{nw} \ast \mathit{brka}(\mathtt{q}+\mathtt{temp}) \ast \mathop{\Large\ast}_{i=0}^{\mathtt{temp}-1} .\, \mathtt{q}+i \mapsto \_ \end{array}\right\}$$

```
// B swallows the block at t. A=B^a still holds because
// the block at t isn't allocated.
```

$$\left\{\begin{array}{l}\exists B.\, \mathit{block}^*\, \mathtt{s}\,\mathtt{q}\, B \ast A = B^{\mathsf{a}} \ast \mathtt{nw} = 1 + \left\lceil \frac{\mathtt{nbytes}}{\mathtt{WORD}} \right\rceil \ast \mathtt{p} = \mathtt{s} \ast \mathtt{temp} > \mathtt{nw} \\ \ast\ \mathit{brka}(\mathtt{q}+\mathtt{temp}) \ast \mathtt{q} \mapsto \_ \ast \mathop{\Large\ast}_{i=\mathtt{q}+1}^{\mathtt{q}+\mathtt{temp}-2} .\, i \mapsto \_ \ast (\mathtt{q}+\mathtt{temp}-1) \mapsto \_ \end{array}\right\}$$

```
t = q->ptr = q+temp-1;
```

$$\left\{\begin{array}{l}\exists B.\, \mathit{block}^*\, \mathtt{s}\,\mathtt{q}\, B \ast A = B^{\mathsf{a}} \ast \mathtt{nw} = 1 + \left\lceil \frac{\mathtt{nbytes}}{\mathtt{WORD}} \right\rceil \ast \mathtt{p} = \mathtt{s} \\ \ast\ \mathit{brka}(\mathtt{t}+1) \ast \mathtt{q} < \mathtt{t} \ast \mathtt{q} \mapsto \mathtt{t} \ast \mathop{\Large\ast}_{i=\mathtt{q}+1}^{\mathtt{t}-1} .\, i \mapsto \_ \ast \mathtt{t} \mapsto \_ \end{array}\right\}$$

$$\left\{\begin{array}{l}\exists B.\, \mathit{block}^*\, \mathtt{s}\,\mathtt{q}\, B \ast A = B^{\mathsf{a}} \ast \mathtt{nw} = 1 + \left\lceil \frac{\mathtt{nbytes}}{\mathtt{WORD}} \right\rceil \ast \mathtt{p} = \mathtt{s} \\ \ast\ \mathit{brka}(\mathtt{t}+1) \ast \mathit{ublock}\,\mathtt{q}\,\mathtt{t}\,\{\mathtt{q}+1 \mapsto_{\mathsf{u}} \mathtt{t}-\mathtt{q}-1\} \ast \mathtt{t} \mapsto \_ \end{array}\right\}$$

```
// B swallows the block at q. A=B^a still holds because
// the block at q isn't allocated.
```

$$\left\{\begin{array}{l}\exists B.\, \mathit{block}^*\, \mathtt{s}\,\mathtt{t}\, B \ast A = B^{\mathsf{a}} \ast \mathtt{nw} = 1 + \left\lceil \frac{\mathtt{nbytes}}{\mathtt{WORD}} \right\rceil \\ \ast\ \mathtt{p} = \mathtt{s} \ast \mathit{brka}(\mathtt{t}+1) \ast \mathtt{t} \mapsto \_ \end{array}\right\}$$

```
t->ptr = setbusy(s);
// reestablish loop inv 1:
```

$$\left\{\begin{array}{l}\exists B_1, B_2.\, \mathit{block}^*\, \mathtt{s}\,\mathtt{p}\, B_1 \ast \mathit{block}^*\, \mathtt{p}\,\mathtt{t}\, B_2 \ast \mathtt{t}_{|1} \mapsto \mathtt{s} \\ \ast\ A = (B_1 \uplus B_2)^{\mathsf{a}} \ast \mathit{brka}(\mathtt{t}+1) \ast \mathtt{nw} = 1 + \left\lceil \frac{\mathtt{nbytes}}{\mathtt{WORD}} \right\rceil \end{array}\right\}$$

```
}
```

$$\left\{\mathsf{false}\right\}$$

```
found:
```

$$\left\{\begin{array}{l}\exists B_1,B_2.\,\mathit{block}^*\,\mathtt{s}\,\mathtt{p}\,B_1 \;*\; \mathit{ublock}\,\mathtt{p}\,\mathtt{q}\,\{\mathtt{p}+1\mapsto_{\mathtt{u}}\mathtt{q}-\mathtt{p}-1\}\\ *\;\mathit{block}^*\,\mathtt{q}\,\mathtt{t}\,B_2 \;*\; \mathtt{t}_{|1}\mapsto\mathtt{s} \;*\; A=(B_1\uplus B_2)^{\mathtt{a}} \;*\; \mathit{brka}(\mathtt{t}+1)\\ *\;\mathtt{nw}=1+\left\lceil\frac{\mathtt{nbytes}}{\mathtt{WORD}}\right\rceil \;*\; \mathtt{q}\geq\mathtt{p}+\mathtt{nw}\end{array}\right\}$$

```
v = p+nw;
```

$$\left\{\begin{array}{l}\exists B_1,B_2.\,\mathit{block}^*\,\mathtt{s}\,\mathtt{p}\,B_1 \;*\; \mathtt{p}<\mathtt{q} \;*\; \mathtt{p}\mapsto\mathtt{q} \;*\; \mathop{\text{✱}}_{i=\mathtt{p}+1}^{\mathtt{q}-1}.\,i\mapsto\_\\ *\;\mathit{block}^*\,\mathtt{q}\,\mathtt{t}\,B_2 \;*\; \mathtt{t}_{|1}\mapsto\mathtt{s} \;*\; A=(B_1\uplus B_2)^{\mathtt{a}} \;*\; \mathit{brka}(\mathtt{t}+1)\\ *\;\mathtt{nw}=1+\left\lceil\frac{\mathtt{nbytes}}{\mathtt{WORD}}\right\rceil \;*\; \mathtt{q}\geq\mathtt{v} \;*\; \mathtt{v}=\mathtt{p}+\mathtt{nw}\end{array}\right\}$$

```
if (q>v) {
```

$$\left\{\begin{array}{l}\exists B_1,B_2.\,\mathit{block}^*\,\mathtt{s}\,\mathtt{p}\,B_1 \;*\; \mathtt{p}<\mathtt{q} \;*\; \mathtt{p}\mapsto\mathtt{q} \;*\; \mathop{\text{✱}}_{i=\mathtt{p}+1}^{\mathtt{q}-1}.\,i\mapsto\_\\ *\;\mathit{block}^*\,\mathtt{q}\,\mathtt{t}\,B_2 \;*\; \mathtt{t}_{|1}\mapsto\mathtt{s} \;*\; A=(B_1\uplus B_2)^{\mathtt{a}} \;*\; \mathit{brka}(\mathtt{t}+1)\\ *\;\mathtt{nw}=1+\left\lceil\frac{\mathtt{nbytes}}{\mathtt{WORD}}\right\rceil \;*\; \mathtt{q}>\mathtt{v} \;*\; \mathtt{v}=\mathtt{p}+\mathtt{nw}\end{array}\right\}$$

```
  v->ptr = p->ptr;
```

$$\left\{\begin{array}{l}\exists B_1,B_2.\,\mathit{block}^*\,\mathtt{s}\,\mathtt{p}\,B_1 \;*\; \mathtt{p}\mapsto\mathtt{q} \;*\; \mathop{\text{✱}}_{i=\mathtt{p}+1}^{\mathtt{v}-1}.\,i\mapsto\_\\ *\;\mathit{ublock}\,\mathtt{v}\,\mathtt{q}\,\{(\mathtt{v}+1)\mapsto_{\mathtt{u}}(\mathtt{q}-\mathtt{v}-1)\}\\ *\;\mathit{block}^*\,\mathtt{q}\,\mathtt{t}\,B_2 \;*\; \mathtt{t}_{|1}\mapsto\mathtt{s} \;*\; A=(B_1\uplus B_2)^{\mathtt{a}} \;*\; \mathit{brka}(\mathtt{t}+1)\\ *\;\mathtt{nw}=1+\left\lceil\frac{\mathtt{nbytes}}{\mathtt{WORD}}\right\rceil \;*\; \mathtt{v}=\mathtt{p}+\mathtt{nw}\end{array}\right\}$$

$$\left\{\begin{array}{l}\exists B_1,B_2.\,\mathit{block}^*\,\mathtt{s}\,\mathtt{p}\,B_1 \;*\; \mathtt{p}\mapsto\mathtt{q} \;*\; \mathop{\text{✱}}_{i=\mathtt{p}+1}^{\mathtt{v}-1}.\,i\mapsto\_ \;*\; \mathit{block}^*\,\mathtt{v}\,\mathtt{t}\,B_2\\ *\;\mathtt{t}_{|1}\mapsto\mathtt{s} \;*\; A=(B_1\uplus B_2)^{\mathtt{a}} \;*\; \mathit{brka}(\mathtt{t}+1) \;*\; \mathtt{nw}=1+\left\lceil\frac{\mathtt{nbytes}}{\mathtt{WORD}}\right\rceil \;*\; \mathtt{v}=\mathtt{p}+\mathtt{nw}\end{array}\right\}$$

```
}
```

$$\left\{\begin{array}{l}\exists B_1,B_2.\,\mathit{block}^*\,\mathtt{s}\,\mathtt{p}\,B_1 \;*\; \mathtt{p}\mapsto\mathtt{q} \;*\; \mathop{\text{✱}}_{i=\mathtt{p}+1}^{\mathtt{v}-1}.\,i\mapsto\_ \;*\; \mathit{block}^*\,\mathtt{v}\,\mathtt{t}\,B_2\\ *\;\mathtt{t}_{|1}\mapsto\mathtt{s} \;*\; A=(B_1\uplus B_2)^{\mathtt{a}} \;*\; \mathit{brka}(\mathtt{t}+1) \;*\; \mathtt{nw}=1+\left\lceil\frac{\mathtt{nbytes}}{\mathtt{WORD}}\right\rceil \;*\; \mathtt{v}=\mathtt{p}+\mathtt{nw}\end{array}\right\}$$

```
p->ptr = setbusy(v);
```

$$\left\{\begin{array}{l}\exists B_1,B_2.\,\mathit{block}^*\,\mathtt{s}\,\mathtt{p}\,B_1 \;*\; \mathtt{p}_{|1}\mapsto\mathtt{v} \;*\; \mathop{\text{✱}}_{i=\mathtt{p}+1}^{\mathtt{v}-1}.\,i\mapsto\_ \;*\; \mathit{block}^*\,\mathtt{v}\,\mathtt{t}\,B_2\\ *\;\mathtt{t}_{|1}\mapsto\mathtt{s} \;*\; A=(B_1\uplus B_2)^{\mathtt{a}} \;*\; \mathit{brka}(\mathtt{t}+1) \;*\; \mathtt{nw}=1+\left\lceil\frac{\mathtt{nbytes}}{\mathtt{WORD}}\right\rceil \;*\; \mathtt{v}=\mathtt{p}+\mathtt{nw}\end{array}\right\}$$

$$\left\{\begin{array}{l}\exists B_1,B_2.\,\mathit{block}^*\,\mathtt{s}\,\mathtt{p}\,B_1 \;*\; \mathit{ablock}\,\mathtt{p}\,\mathtt{v}\,\{\mathtt{p}+1\mapsto_{\mathtt{a}}\mathtt{nw}-1\} \;*\; \mathit{block}^*\,\mathtt{v}\,\mathtt{t}\,B_2\\ *\;\mathtt{t}_{|1}\mapsto\mathtt{s} \;*\; A=(B_1\uplus B_2)^{\mathtt{a}} \;*\; \mathit{brka}(\mathtt{t}+1) \;*\; \mathtt{nw}=1+\left\lceil\frac{\mathtt{nbytes}}{\mathtt{WORD}}\right\rceil\\ *\;\mathtt{p}_{|1}\overset{.5}{\mapsto}\mathtt{v} \;*\; \mathop{\text{✱}}_{i=\mathtt{p}+1}^{\mathtt{v}-1}.\,i\mapsto\_ \;*\; \mathtt{v}=\mathtt{p}+\mathtt{nw}\end{array}\right\}$$

```
// use lemma to deduce that B1 and p+1 are disjoint
```

$$\left\{\begin{array}{l}\exists B_1,B_2.\,\mathit{block}^*\,\mathtt{s}\,\mathtt{v}\,B_1 \;*\; \mathit{block}^*\,\mathtt{v}\,\mathtt{t}\,B_2 \;*\; \mathtt{t}_{|1}\mapsto\mathtt{s}\\ *\;A\uplus\{\mathtt{p}+1\mapsto\left\lceil\frac{\mathtt{nbytes}}{\mathtt{WORD}}\right\rceil\}=(B_1\uplus B_2)^{\mathtt{a}}\\ *\;\mathit{brka}(\mathtt{t}+1) \;*\; \mathtt{p}_{|1}\overset{.5}{\mapsto}\mathtt{p}+\left\lceil\frac{\mathtt{nbytes}}{\mathtt{WORD}}\right\rceil+1\\ *\;\mathop{\text{✱}}_{i=0}^{\lceil\mathtt{nbytes}/\mathtt{WORD}\rceil-1}.\,\mathtt{p}+1+i\mapsto\_\end{array}\right\}$$

$$\left\{\begin{array}{l}(\mathit{arena}(A\uplus\{\mathtt{ret}\mapsto\left\lceil\frac{\mathtt{nbytes}}{\mathtt{WORD}}\right\rceil\}) \;*\; \mathop{\text{✱}}_{i=0}^{\lceil\mathtt{nbytes}/\mathtt{WORD}\rceil-1}.\,\mathtt{ret}+i\mapsto\_\\ \;*\;(\mathtt{ret}-1)_{|1}\overset{.5}{\mapsto}\mathtt{ret}+\left\lceil\frac{\mathtt{nbytes}}{\mathtt{WORD}}\right\rceil)[\mathtt{p}+1/\mathtt{ret}]\end{array}\right\}$$

```
return((char *)(p+1));
```

$$\left\{\mathsf{false}\right\}$$

}

$$\left\{\begin{pmatrix} arena(A \uplus \{\mathtt{ret} \mapsto \lceil \mathtt{nbytes}/\mathtt{WORD}\rceil\}) \\ * \ \bigast_{i=0}^{\lceil \mathtt{nbytes}/\mathtt{WORD}\rceil-1}.\, \mathtt{ret}+i \mapsto \_ \\ * \ (\mathtt{ret}-1)_{|1} \overset{.5}{\mapsto} \mathtt{ret}+\lceil \mathtt{nbytes}/\mathtt{WORD}\rceil \end{pmatrix} \vee (arena\ A \ * \ \mathtt{ret}=0)\right\}$$

```
// end region update
```

$$\left\{\begin{pmatrix} \boxed{arena(A \uplus \{\mathtt{ret} \mapsto \lceil \mathtt{nbytes}/\mathtt{WORD}\rceil\})} \\ * \ \bigast_{i=0}^{\lceil \mathtt{nbytes}/\mathtt{WORD}\rceil-1}.\, \mathtt{ret}+i \mapsto \_ \\ * \ (\mathtt{ret}-1)_{|1} \overset{.5}{\mapsto} \mathtt{ret}+\lceil \mathtt{nbytes}/\mathtt{WORD}\rceil \end{pmatrix} \vee (\boxed{arena\ A} \ * \ \mathtt{ret}=0)\right\}$$

```
// end existential
```

$$\left\{\begin{pmatrix} \boxed{\exists A.\, arena(A \uplus \{\mathtt{ret} \mapsto \lceil \mathtt{nbytes}/\mathtt{WORD}\rceil\})} \\ * \ \bigast_{i=0}^{\lceil \mathtt{nbytes}/\mathtt{WORD}\rceil-1}.\, \mathtt{ret}+i \mapsto \_ \\ * \ (\mathtt{ret}-1)_{|1} \overset{.5}{\mapsto} \mathtt{ret}+\lceil \mathtt{nbytes}/\mathtt{WORD}\rceil \end{pmatrix} \vee (\boxed{\exists A.\, arena\ A} \ * \ \mathtt{ret}=0)\right\}$$

// note that $\exists A.\, arena(A \uplus \{\mathtt{ret} \mapsto \lceil \mathtt{nbytes}/\mathtt{WORD}\rceil\})$ implies $\exists A.\, arena(A)$

$$\left\{\begin{pmatrix} \boxed{\exists A.\, arena\ A} \\ * \ \boxed{\exists A.\, arena(A \uplus \{\mathtt{ret} \mapsto \lceil \mathtt{nbytes}/\mathtt{WORD}\rceil\})} \\ * \ \bigast_{i=0}^{\lceil \mathtt{nbytes}/\mathtt{WORD}\rceil-1}.\, \mathtt{ret}+i \mapsto \_ \\ * \ (\mathtt{ret}-1)_{|1} \overset{.5}{\mapsto} \mathtt{ret}+\lceil \mathtt{nbytes}/\mathtt{WORD}\rceil \end{pmatrix} \vee (\boxed{\exists A.\, arena\ A} \ * \ \mathtt{ret}=0)\right\}$$

$$\left\{\begin{pmatrix} \boxed{\exists A.\, uninit\, A \ \vee \ arena\ A} \\ * \ \boxed{\exists A.\, arena(A \uplus \{\mathtt{ret} \mapsto \lceil \mathtt{nbytes}/\mathtt{WORD}\rceil\})} \\ * \ \bigast_{i=0}^{\lceil \mathtt{nbytes}/\mathtt{WORD}\rceil-1}.\, \mathtt{ret}+i \mapsto \_ \\ * \ (\mathtt{ret}-1)_{|1} \overset{.5}{\mapsto} \mathtt{ret}+\lceil \mathtt{nbytes}/\mathtt{WORD}\rceil \end{pmatrix} \vee (\boxed{\exists A.\, uninit\, A \ \vee \ arena\ A} \ * \ \mathtt{ret}=0)\right\}$$

$$\left\{\begin{pmatrix} anArena \\ * \ token(\mathtt{ret}, \lceil \mathtt{nbytes}/\mathtt{WORD}\rceil) \\ * \ \bigast_{i=0}^{\lceil \mathtt{nbytes}/\mathtt{WORD}\rceil-1}.\, \mathtt{ret}+i \mapsto \_ \end{pmatrix} \vee (anArena \ * \ \mathtt{ret}=0)\right\}$$

$$\left\{anArena \ * \ ((token\, \mathtt{ret}\ \lceil \mathtt{nbytes}/\mathtt{WORD}\rceil \ * \ \bigast_{i=0}^{\lceil \mathtt{nbytes}/\mathtt{WORD}\rceil-1}.\, \mathtt{ret}+i \mapsto \_) \ \vee \ \mathtt{ret}=0)\right\}$$

Verification of free routine

```
free(register char *ap)
```

$$\left\{anArena \ * \ \exists n.\, token\ \mathsf{ap}\, n \ * \ \bigast_{i=0}^{n-1}.\, (\mathsf{ap}+i) \mapsto \_\right\}$$

$$\left\{\begin{array}{l}\exists n.\, \boxed{\exists A.\, uninit\, A \ \vee \ arena\ A} \ * \ \boxed{\exists A.\, arena(A \uplus \{\mathsf{ap} \mapsto n\})} \ * \ (\mathsf{ap}-1)_{|1} \overset{.5}{\mapsto} (\mathsf{ap}+n) \\ * \ \bigast_{i=0}^{n-1}.\, (\mathsf{ap}+i) \mapsto \_\end{array}\right\}$$

$$\left\{\exists n.\, \boxed{\exists A.\, arena(A \uplus \{\mathsf{ap} \mapsto n\})} \ * \ (\mathsf{ap}-1)_{|1} \overset{.5}{\mapsto} (\mathsf{ap}+n) \ * \ \bigast_{i=0}^{n-1}.\, \mathsf{ap}+i \mapsto \_\right\}$$

```
//begin existential
```

$$\left\{\boxed{\mathit{arena}(A \uplus \{\mathtt{ap} \mapsto n\})} \;*\; (\mathtt{ap}-1)_{|1} \overset{.5}{\mapsto} (\mathtt{ap}+n) \;*\; \mathop{\ast}_{i=0}^{n-1}.\, \mathtt{ap}+i \mapsto \_\right\}$$

```
//begin "Free x" action
{
```

$$\left\{\mathit{arena}(A \uplus \{\mathtt{ap} \mapsto n\}) \;*\; (\mathtt{ap}-1)_{|1} \overset{.5}{\mapsto} (\mathtt{ap}+n) \;*\; \mathop{\ast}_{i=0}^{n-1}.\, \mathtt{ap}+i \mapsto \_\right\}$$

$$\left\{\begin{array}{l}\exists B_1, B_2.\, \mathit{block}^*\ \mathtt{s}\,\mathtt{v}\, B_1 \;*\; \mathit{block}^*\ \mathtt{v}\,\mathtt{t}\, B_2 \;*\; A \uplus \{\mathtt{ap} \mapsto n\} = (B_1 \uplus B_2)^{\mathsf{a}} \;*\; \mathtt{t}_{|1} \mapsto \mathtt{s} \\ *\; \mathit{brka}(\mathtt{t}+1) \;*\; (\mathtt{ap}-1)_{|1} \overset{.5}{\mapsto} (\mathtt{ap}+n) \;*\; \mathop{\ast}_{i=0}^{n-1}.\, \mathtt{ap}+i \mapsto \_\end{array}\right\}$$

```
// use lemma to deduce that B1 and B2 are disjoint
```

$$\left\{\begin{array}{l}\exists B.\, \mathit{block}^*\ \mathtt{s}\,\mathtt{t}\, B \;*\; A \uplus \{\mathtt{ap} \mapsto n\} = B^{\mathsf{a}} \;*\; \mathtt{t}_{|1} \mapsto \mathtt{s} \\ *\; \mathit{brka}(\mathtt{t}+1) \;*\; (\mathtt{ap}-1)_{|1} \overset{.5}{\mapsto} (\mathtt{ap}+n) \;*\; \mathop{\ast}_{i=0}^{n-1}.\, \mathtt{ap}+i \mapsto \_\end{array}\right\}$$

// note that $\{x \mapsto_{\mathsf{a}} n\} \in B$ implies $\exists B_1, B_2.\, B = B_1 \uplus \{x \mapsto_{\mathsf{a}} n\} \uplus B_2$

$$\left\{\begin{array}{l}\exists B_1, B_2.\, \mathit{block}^*\ \mathtt{s}\,(\mathtt{ap}-1)\, B_1 \;*\; \mathit{ablock}\,(\mathtt{ap}-1)\,(\mathtt{ap}+n)\,\{\mathtt{ap} \mapsto_{\mathsf{a}} n\} \\ *\; \mathit{block}^*\ (\mathtt{ap}+n)\,\mathtt{t}\, B_2 \;*\; A \uplus \{\mathtt{ap} \mapsto n\} = (B_1 \uplus \{\mathtt{ap} \mapsto_{\mathsf{a}} n\} \uplus B_2)^{\mathsf{a}} \;*\; \mathtt{t}_{|1} \mapsto \mathtt{s} \\ *\; \mathit{brka}(\mathtt{t}+1) \;*\; (\mathtt{ap}-1)_{|1} \overset{.5}{\mapsto} (\mathtt{ap}+n) \;*\; \mathop{\ast}_{i=0}^{n-1}.\, \mathtt{ap}+i \mapsto \_\end{array}\right\}$$

// by cancellativity of $\uplus$:

$$\left\{\begin{array}{l}\exists B_1, B_2.\, \mathit{block}^*\ \mathtt{s}\,(\mathtt{ap}-1)\, B_1 \;*\; \mathit{ablock}\,(\mathtt{ap}-1)\,(\mathtt{ap}+n)\,\{\mathtt{ap} \mapsto_{\mathsf{a}} n\} \\ *\; \mathit{block}^*\ (\mathtt{ap}+n)\,\mathtt{t}\, B_2 \;*\; A = (B_1 \uplus B_2)^{\mathsf{a}} \;*\; \mathtt{t}_{|1} \mapsto \mathtt{s} \\ *\; \mathit{brka}(\mathtt{t}+1) \;*\; (\mathtt{ap}-1)_{|1} \overset{.5}{\mapsto} (\mathtt{ap}+n) \;*\; \mathop{\ast}_{i=0}^{n-1}.\, \mathtt{ap}+i \mapsto \_\end{array}\right\}$$

```
register struct store *p = (struct store *)ap;
v = --p;
```

$$\left\{\begin{array}{l}\exists B_1, B_2.\, \mathit{block}^*\ \mathtt{s}\,\mathtt{p}\, B_1 \;*\; \mathit{ablock}\,\mathtt{p}\,(\mathtt{p}+1+n)\,\{\mathtt{p}+1 \mapsto_{\mathsf{a}} n\} \\ *\; \mathit{block}^*\ (\mathtt{p}+1+n)\,\mathtt{t}\, B_2 \;*\; A = (B_1 \uplus B_2)^{\mathsf{a}} \;*\; \mathtt{t}_{|1} \mapsto \mathtt{s} \\ *\; \mathit{brka}(\mathtt{t}+1) \;*\; \mathtt{p}_{|1} \overset{.5}{\mapsto} (\mathtt{p}+1+n) \;*\; \mathop{\ast}_{i=0}^{n-1}.\, \mathtt{p}+1+i \mapsto \_ \;*\; \mathtt{p} = \mathtt{v}\end{array}\right\}$$

$$\left\{\begin{array}{l}\exists B_1, B_2.\, \mathit{block}^*\ \mathtt{s}\,\mathtt{p}\, B_1 \;*\; \mathtt{p}_{|1} \mapsto \mathtt{p}+1+n \;*\; \mathop{\ast}_{i=0}^{n-1}.\, \mathtt{p}+1+i \mapsto \_ \\ *\; \mathit{block}^*\ (\mathtt{p}+1+n)\,\mathtt{t}\, B_2 \;*\; A = (B_1 \uplus B_2)^{\mathsf{a}} \;*\; \mathtt{t}_{|1} \mapsto \mathtt{s} \;*\; \mathit{brka}(\mathtt{t}+1) \;*\; \mathtt{p} = \mathtt{v}\end{array}\right\}$$

```
p->ptr = clearbusy(p->ptr);
```

$$\left\{\begin{array}{l}\exists B_1, B_2.\, \mathit{block}^*\ \mathtt{s}\,\mathtt{p}\, B_1 \;*\; \mathtt{p} \mapsto \mathtt{p}+1+n \;*\; \mathop{\ast}_{i=0}^{n-1}.\, \mathtt{p}+1+i \mapsto \_ \\ *\; \mathit{block}^*\ (\mathtt{p}+1+n)\,\mathtt{t}\, B_2 \;*\; A = (B_1 \uplus B_2)^{\mathsf{a}} \;*\; \mathtt{t}_{|1} \mapsto \mathtt{s} \;*\; \mathit{brka}(\mathtt{t}+1) \;*\; \mathtt{p} = \mathtt{v}\end{array}\right\}$$

$$\left\{\begin{array}{l}\exists B_1, B_2.\, \mathit{block}^*\ \mathtt{s}\,\mathtt{p}\, B_1 \;*\; \mathit{ublock}\,\mathtt{p}\,(\mathtt{p}+1+n)\,\{\mathtt{p}+1 \mapsto_{\mathsf{u}} n\} \\ *\; \mathit{block}^*\ (\mathtt{p}+1+n)\,\mathtt{t}\, B_2 \;*\; A = (B_1 \uplus B_2)^{\mathsf{a}} \;*\; \mathtt{t}_{|1} \mapsto \mathtt{s} \;*\; \mathit{brka}(\mathtt{t}+1) \;*\; \mathtt{p} = \mathtt{v}\end{array}\right\}$$

```
// use lemma to deduce that p and B2 are disjoint
```

$$\left\{\exists B_1, B_2.\, \mathit{block}^*\ \mathtt{s}\,\mathtt{v}\, B_1 \;*\; \mathit{block}^*\ \mathtt{v}\,\mathtt{t}\, B_2 \;*\; A = (B_1 \uplus B_2)^{\mathsf{a}} \;*\; \mathtt{t}_{|1} \mapsto \mathtt{s} \;*\; \mathit{brka}(\mathtt{t}+1)\right\}$$

$$\left\{\mathit{arena}\, A\right\}$$

```
}
//end "Free x" action
```

$\left\{\boxed{arena\,A}\right\}$

```
//end existential
```

$\left\{\boxed{\exists A.\, arena\,A}\right\}$

$\left\{\boxed{\exists A.\, uninit\,A \;\vee\; arena\,A}\right\}$

$\left\{anArena\right\}$