```prolog
belong( X, cons(X,L) ).

belong( X, cons(Y,L) ) :- belong( X, L ).


shared( X, L1, L2 ) :- belong( X, L1 ), belong( X, L2 ).

chosen( X, L1, L2 ) :- shared( X, L1, L2 ), p( X ).


p( Y ) :- q( Y ).

q( 3 ).



?- chosen( Z, cons(1,cons(2,cons(3,null))),
           cons(1,cons(3,cons(5,null))) ).
```

```
belong( X, cons(X,L) ).

belong( X, cons(Y,L) ) :- belong( X, L ).

shared( X, L1, L2 ) :- belong( X, L1 ), belong( X, L2 ).

chosen( X, L1, L2 ) :- shared( X, L1, L2 ), p( X ).

p( Y ) :- q( Y ).

q( 3 ).

?- chosen( Z, cons(1,cons(2,cons(3,null))),
              cons(1,cons(3,cons(5,null))) ).
```

```prolog
belong( X, cons(X,L) ).

belong( X, cons(Y,L) ) :- belong( X, L ).

shared( X, L1, L2 ) :- belong( X, L1 ), belong( X, L2 ).

chosen( X, L1, L2 ) :- shared( X, L1, L2 ), p( X ).

p( Y ) :- q( Y ).

q( 3 ).

?- chosen( Z, cons(1,cons(2,cons(3,null))),
           cons(1,cons(3,cons(5,null))) ).
```

```
belong( X, cons(X,L) ).

belong( X, cons(Y,L) ) :- belong( X, L ).

shared( X, L1, L2 ) :- belong( X, L1 ), belong( X, L2 ).
chosen( X, L1, L2 ) :- shared( X, L1, L2 ), p( X ).

p( Y ) :- q( Y ).

q( 3 ).

?- chosen( Z, cons(1,cons(2,cons(3,null))),
              cons(1,cons(3,cons(5,null))) ).
```
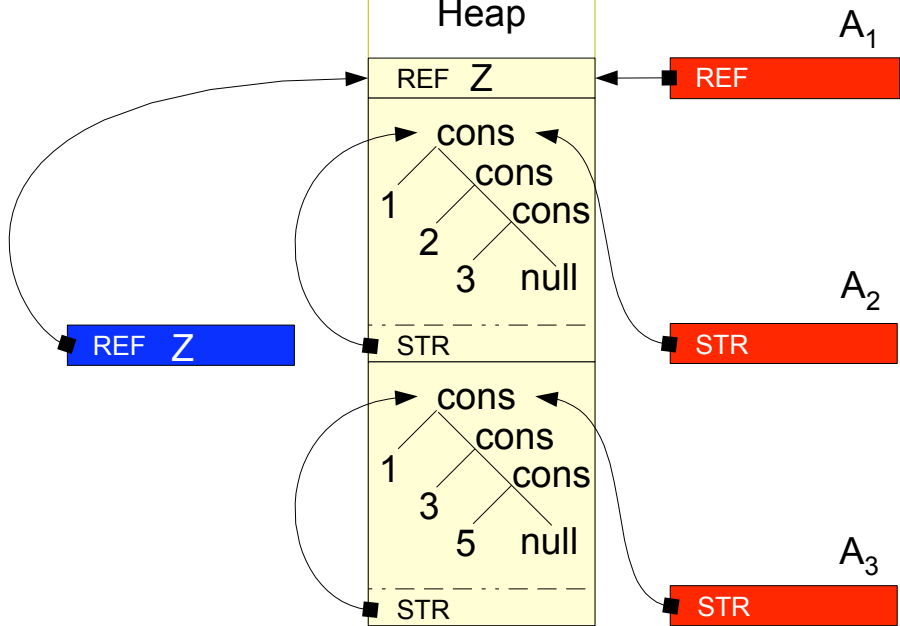
```
belong( X, cons(X,L) ).

belong( X, cons(Y,L) ) :- belong( X, L ).


shared( X, L1, L2 ) :- belong( X, L1 ), belong( X, L2 ).

chosen( X, L1, L2 ) :- shared( X, L1, L2 ), p( X ).


p( Y ) :- q( Y ).

q( 3 ).


?- chosen( Z, cons(1,cons(2,cons(3,null))),
              cons(1,cons(3,cons(5,null))) ).
```
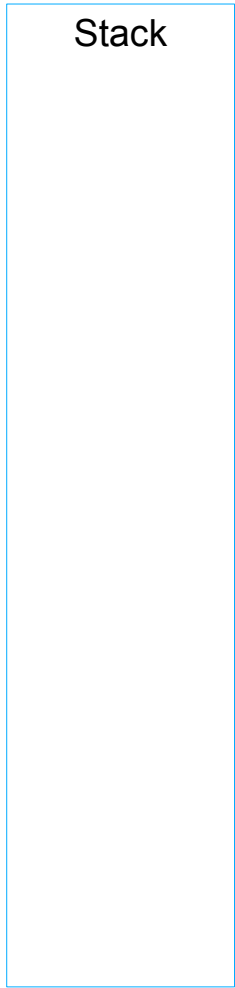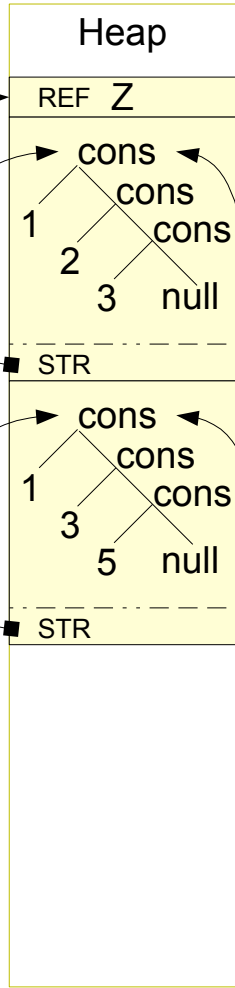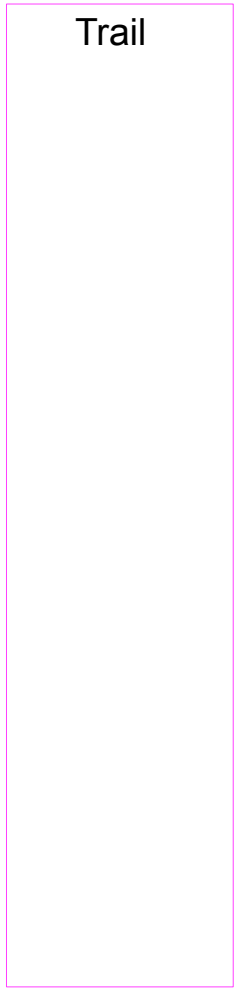
```prolog
belong( X, cons(X,L) ).
belong( X, cons(Y,L) ) :- belong( X, L ).

shared( X, L1, L2 ) :- belong( X, L1 ), belong( X, L2 ).
chosen( X, L1, L2 ) :- shared( X, L1, L2 ), p( X ).

p( Y ) :- q( Y ).

q( 3 ).

?- chosen( Z, cons(1,cons(2,cons(3,null))),
              cons(1,cons(3,cons(5,null))) ).
```
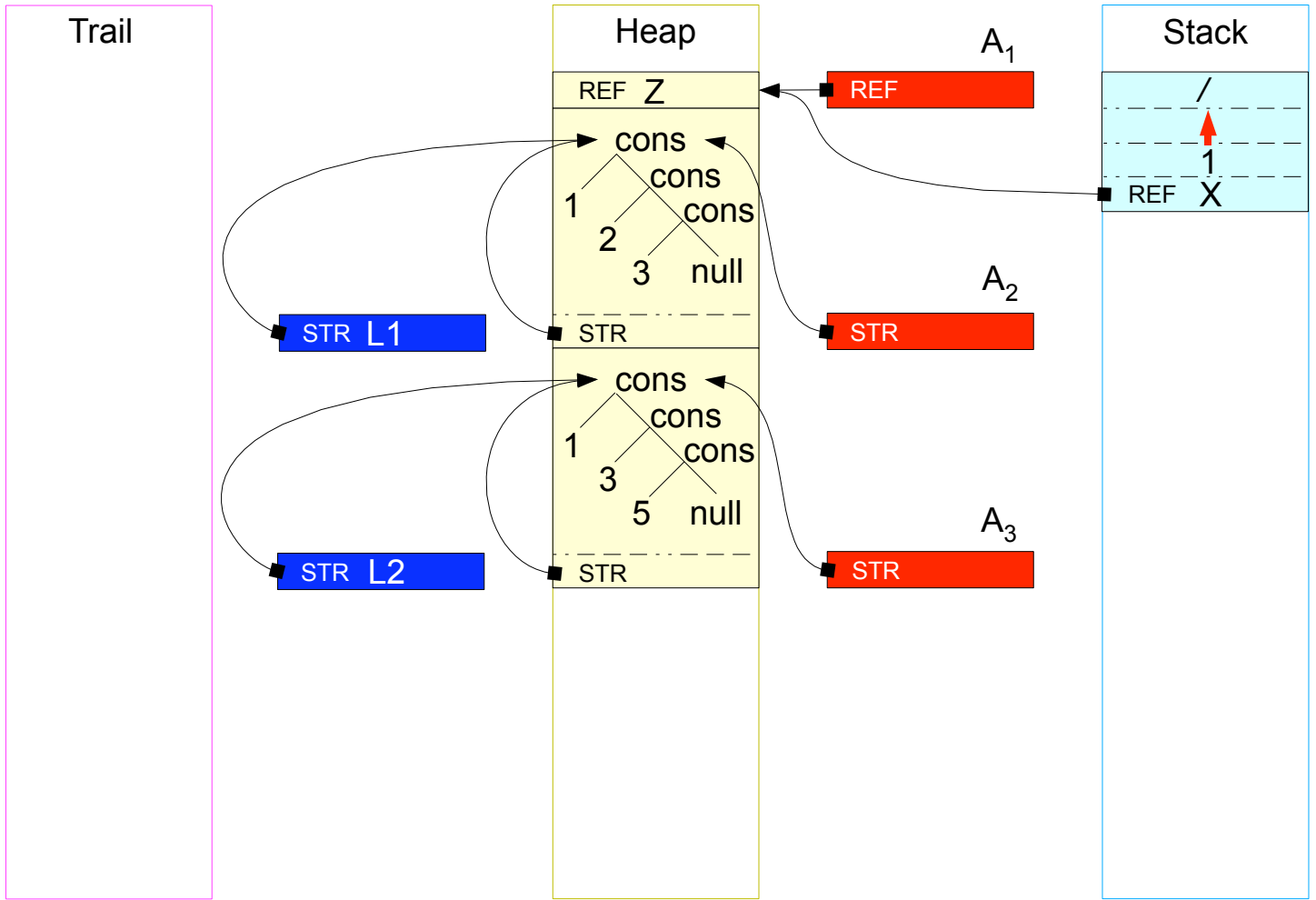
```
?- chosen( Z, [1,2,3], [1,3,5] ).

?- shared( Z, [1,2,3], [1,3,5] ), p( Z ).

?- belong( Z, [1,2,3] ), belong( Z, [1,3,5] ), p( Z ).
          {Z=1}
?- belong( 1, [1,3,5] ), p( 1 ).

?- p( 1 ).

?- q( 1 ).

   fail
```

```
?- belong( Z, [2,3] ), belong( Z, [1,3,5] ), p( Z ).
               {Z=2}
?- belong( 2, [1,3,5] ), p( 2 ).
        .
        .
        .
     fail
```

```
?- belong( 1, [3,5] ), p( 1 ).
        .
        .
        .
     fail
```

```
?- belong( Z, [3] ), belong( Z, [1,3,5] ), p( Z ).
             {Z=3}
?- belong( 3, [1,3,5] ), p( 3 ).

?- belong( 3, [3,5] ), p( 3 ).

?- p( 3 ).

?- q( 3 ).

   succeed
```

```
?- belong( Z, [] ), ...
        ...
     fail
```

```
belong( X, cons(X,L) ).
belong( X, cons(Y,L) ) :- belong( X, L ).

shared( X, L1, L2 ) :- belong( X, L1 ), belong( X, L2 ).
chosen( X, L1, L2 ) :- shared( X, L1, L2 ), p( X ).

p( Y ) :- q( Y ).

q( 3 ).

?- chosen( Z, cons(1,cons(2,cons(3,null))),
              cons(1,cons(3,cons(5,null))) ).
```

```
belong( X, cons(X,L) ).

belong( X, cons(Y,L) ) :- belong( X, L ).

shared( X, L1, L2 ) :- belong( X, L1 ), belong( X, L2 ).

chosen( X, L1, L2 ) :- shared( X, L1, L2 ), p( X ).

p( Y ) :- q( Y ).

q( 3 ).

?- chosen( Z, cons(1,cons(2,cons(3,null))),
           cons(1,cons(3,cons(5,null))) ).
```
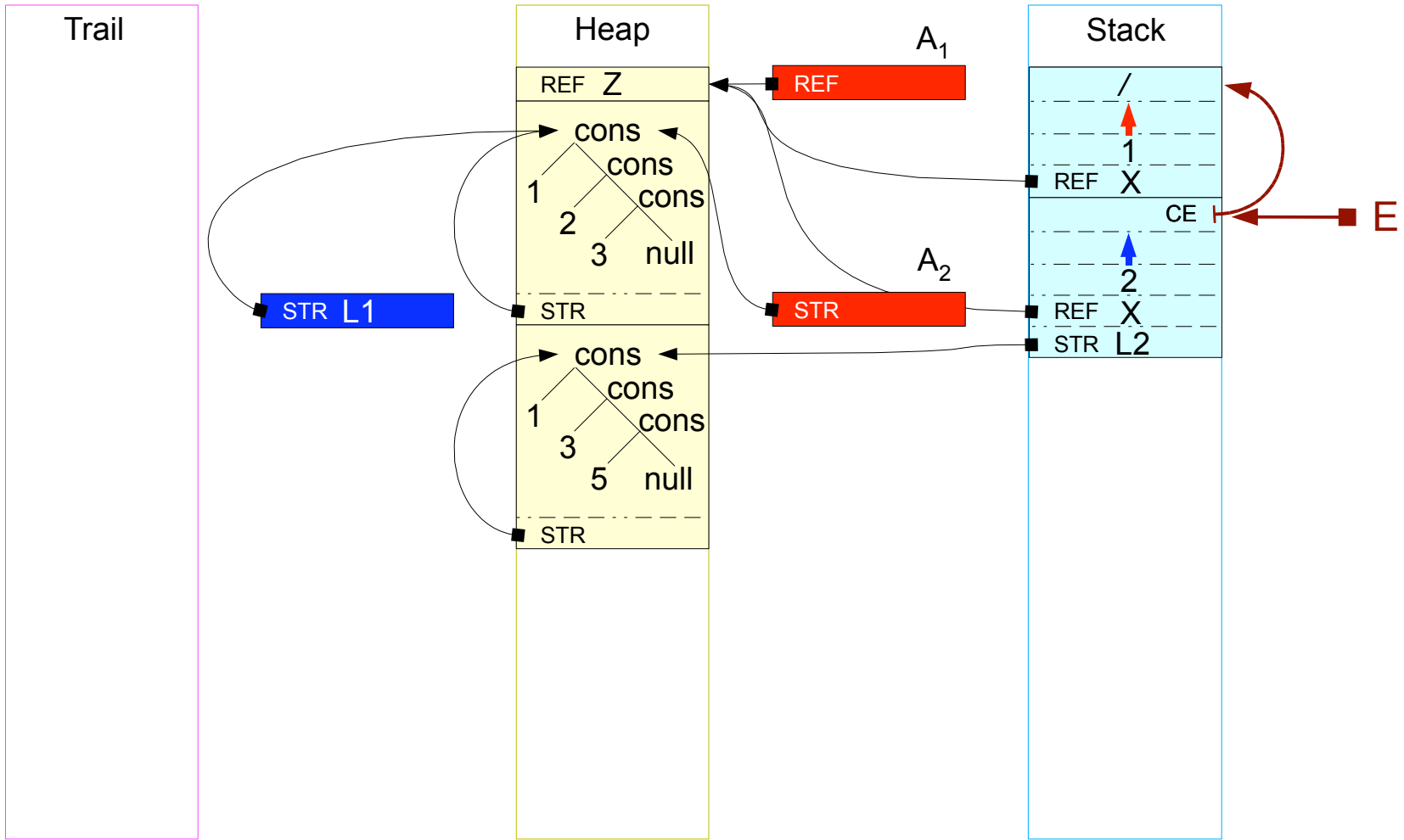
```
?- chosen( Z, [1,2,3], [1,3,5] ).
        |
?- shared( Z, [1,2,3], [1,3,5] ), p( Z ).
        |
?- belong( Z, [1,2,3] ), belong( Z, [1,3,5] ), p( Z ).
        {Z=1}
?- belong( 1, [1,3,5] ), p( 1 ).

?- p( 1 ).

?- q( 1 ).

   fail
```

```
?- belong( Z, [2,3] ), belong( Z, [1,3,5] ), p( Z ).
              {Z=2}
?- belong( 2, [1,3,5] ), p( 2 ).
        .
        .
        .
      fail
```

```
?- belong( 1, [3,5] ), p( 1 ).
        .
        .
        .
      fail
```

```
?- belong( Z, [3] ), belong( Z, [1,3,5] ), p( Z ).
            {Z=3}
?- belong( 3, [1,3,5] ), p( 3 ).
        |
?- belong( 3, [3,5] ), p( 3 ).
        |
?- p( 3 ).
        |
?- q( 3 ).
        |
   succeed
```
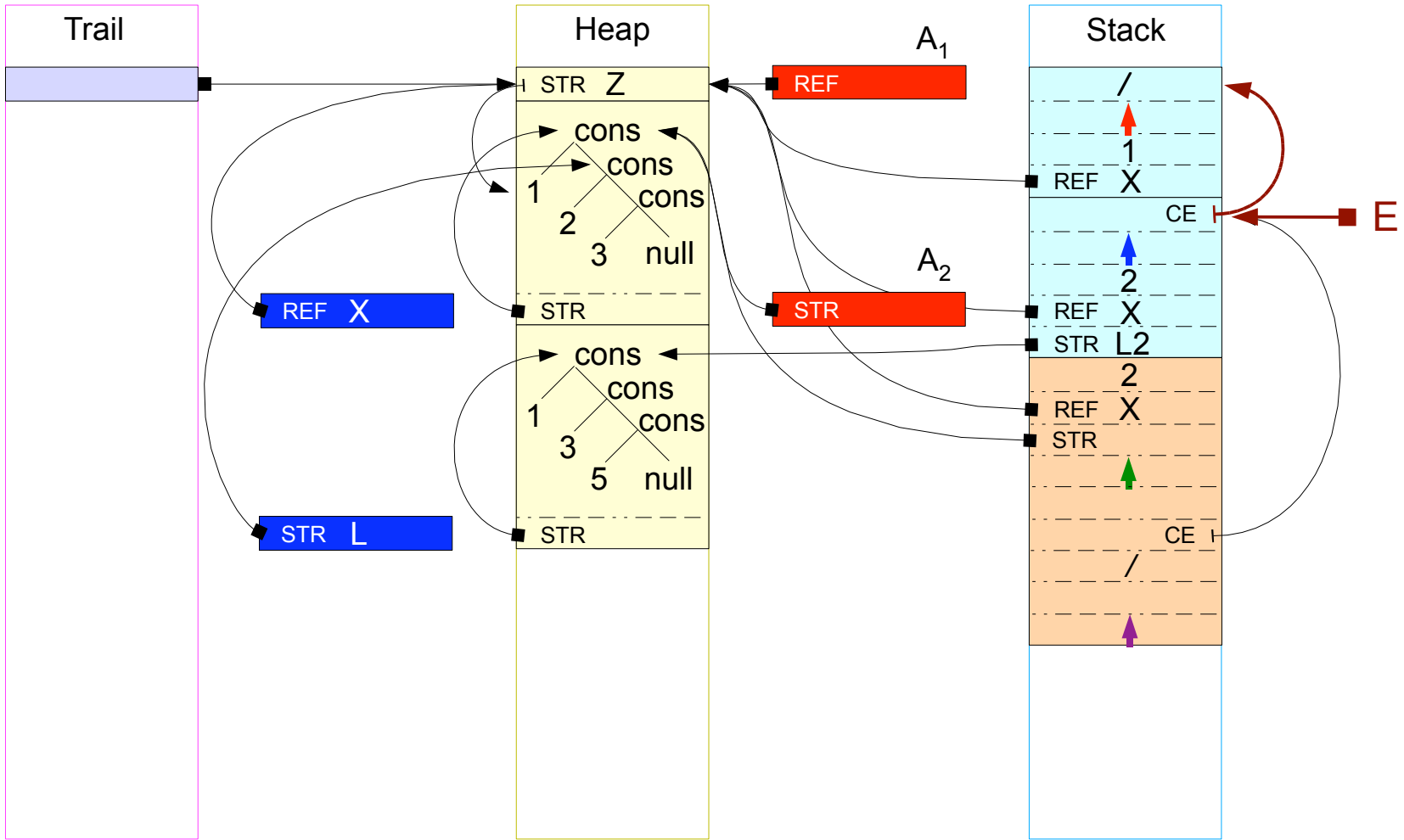
```
?- belong( Z, [] ), ...
        ...
      fail
```
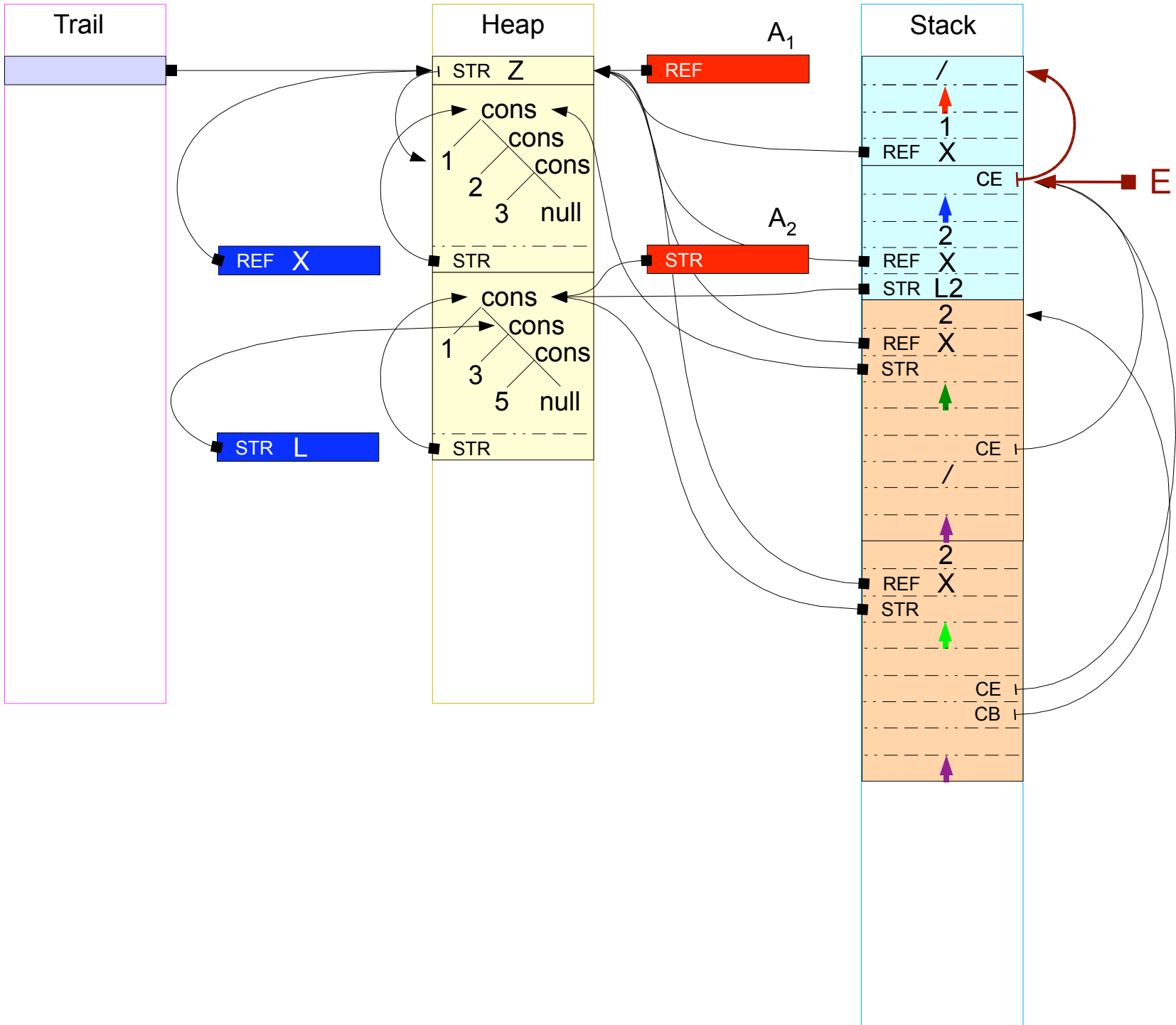
```
belong( X, cons(X,L) ).
belong( X, cons(Y,L) ) :- belong( X, L ).

shared( X, L1, L2 ) :- belong( X, L1 ), belong( X, L2 ).
chosen( X, L1, L2 ) :- shared( X, L1, L2 ), p( X ).

p( Y ) :- q( Y ).

q( 3 ).

?- chosen( Z, cons(1,cons(2,cons(3,null))),
            cons(1,cons(3,cons(5,null))) ).
```

```
?- chosen( Z, [1,2,3], [1,3,5] ).
           |
?- shared( Z, [1,2,3], [1,3,5] ), p( Z ).
           |
?- belong( Z, [1,2,3] ), belong( Z, [1,3,5] ), p( Z ).
       {Z=1}
?- belong( 1, [1,3,5] ), p( 1 ).

?- p( 1 ).
     |
?- q( 1 ).
     |
   fail

      ?- belong( 1, [3,5] ), p( 1 ).
              .
              .
              .
            fail
```

```
?- belong( Z, [2,3] ), belong( Z, [1,3,5] ), p( Z ).
           {Z=2}
?- belong( 2, [1,3,5] ), p( 2 ).
        .
        .
        .
      fail
```

```
?- belong( Z, [3] ), belong( Z, [1,3,5] ), p( Z ).
          {Z=3}
?- belong( 3, [1,3,5] ), p( 3 ).
       |
?- belong( 3, [3,5] ), p( 3 ).
       |
?- p( 3 ).
       |
?- q( 3 ).
       |
   succeed

              ?- belong( Z, [] ), ...
                      ...
                    fail
```

Trail

Heap

A_1

REF

Stack

REF Z

cons

cons

cons

1

2

3 null

STR

REF X

CE

A_2

STR

REF X

STR L2

cons

cons

cons

1

3

5 null

STR

REF X

STR

CE

E

/

```
belong( X, cons(X,L) ).
belong( X, cons(Y,L) ) :- belong( X, L ).

shared( X, L1, L2 ) :- belong( X, L1 ), belong( X, L2 ).
chosen( X, L1, L2 ) :- shared( X, L1, L2 ), p( X ).

p( Y ) :- q( Y ).

q( 3 ).

?- chosen( Z, cons(1,cons(2,cons(3,null))),
            cons(1,cons(3,cons(5,null))) ).
```

```
?- chosen( Z, [1,2,3], [1,3,5] ).

?- shared( Z, [1,2,3], [1,3,5] ), p( Z ).

?- belong( Z, [1,2,3] ), belong( Z, [1,3,5] ), p( Z ).
        {Z=1}
?- belong( 1, [1,3,5] ), p( 1 ).                    ?- belong( Z, [2,3] ), belong( Z, [1,3,5] ), p( Z ).
                                                              {Z=2}
?- p( 1 ).                                          ?- belong( 2, [1,3,5] ), p( 2 ).
   |                                                        .
?- q( 1 ).                                                  .
   |                                                        .
  fail                                                    fail

        ?- belong( 1, [3,5] ), p( 1 ).                              ?- belong( Z, [3] ), belong( Z, [1,3,5] ), p( Z ).
               .                                                            {Z=3}
               .                                                    ?- belong( 3, [1,3,5] ), p( 3 ).
               .                                                            |
             fail                                                   ?- belong( 3, [3,5] ), p( 3 ).
                                                                            |
                                                                    ?- p( 3 ).
                                                                            |
                                                                    ?- q( 3 ).                    ?- belong( Z, [] ), ...
                                                                            |                            ...
                                                                         succeed                        fail
```
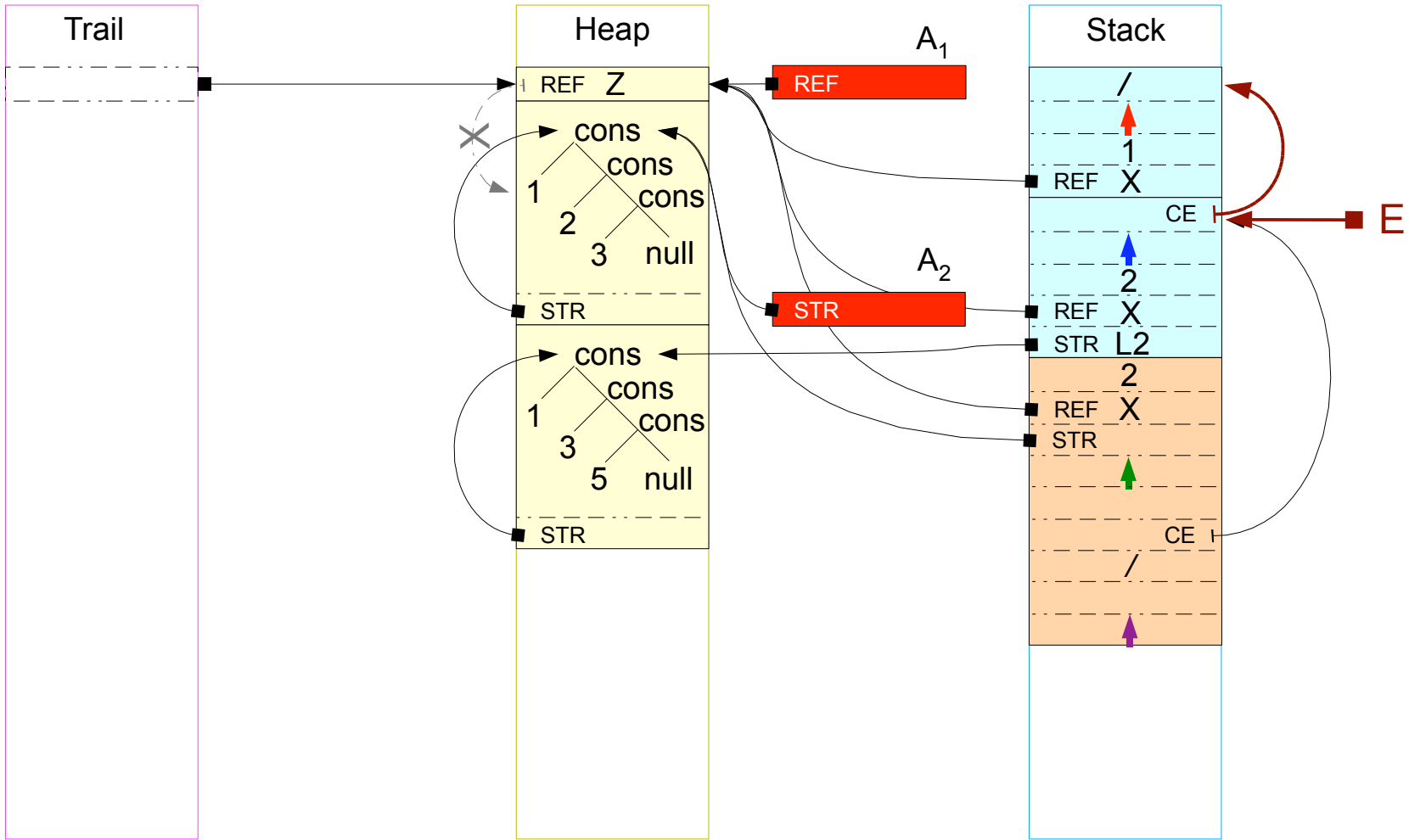
```
belong( X, cons(X,L) ).

belong( X, cons(Y,L) ) :- belong( X, L ).

shared( X, L1, L2 ) :- belong( X, L1 ), belong( X, L2 ).

chosen( X, L1, L2 ) :- shared( X, L1, L2 ), p( X ).

p( Y ) :- q( Y ).

q( 3 ).

?- chosen( Z, cons(1,cons(2,cons(3,null))),
              cons(1,cons(3,cons(5,null))) ).
```

```
?- chosen( Z, [1,2,3], [1,3,5] ).

?- shared( Z, [1,2,3], [1,3,5] ), p( Z ).

?- belong( Z, [1,2,3] ), belong( Z, [1,3,5] ), p( Z ).
         {Z=1}

?- belong( 1, [1,3,5] ), p( 1 ).                    ?- belong( Z, [2,3] ), belong( Z, [1,3,5] ), p( Z ).
                                                                  {Z=2}
?- p( 1 ).                                          ?- belong( 2, [1,3,5] ), p( 2 ).
                                                              .
?- q( 1 ).                                                    .
                                                              .
    fail                                                    fail


         ?- belong( 1, [3,5] ), p( 1 ).
                     .                              ?- belong( Z, [3] ), belong( Z, [1,3,5] ), p( Z ).
                     .                                        {Z=3}
                     .                              ?- belong( 3, [1,3,5] ), p( 3 ).
                   fail
                                                    ?- belong( 3, [3,5] ), p( 3 ).

                                                    ?- p( 3 ).

                                                    ?- q( 3 ).                  ?- belong( Z, [] ), ...
                                                                                        ...
                                                      succeed                          fail
```
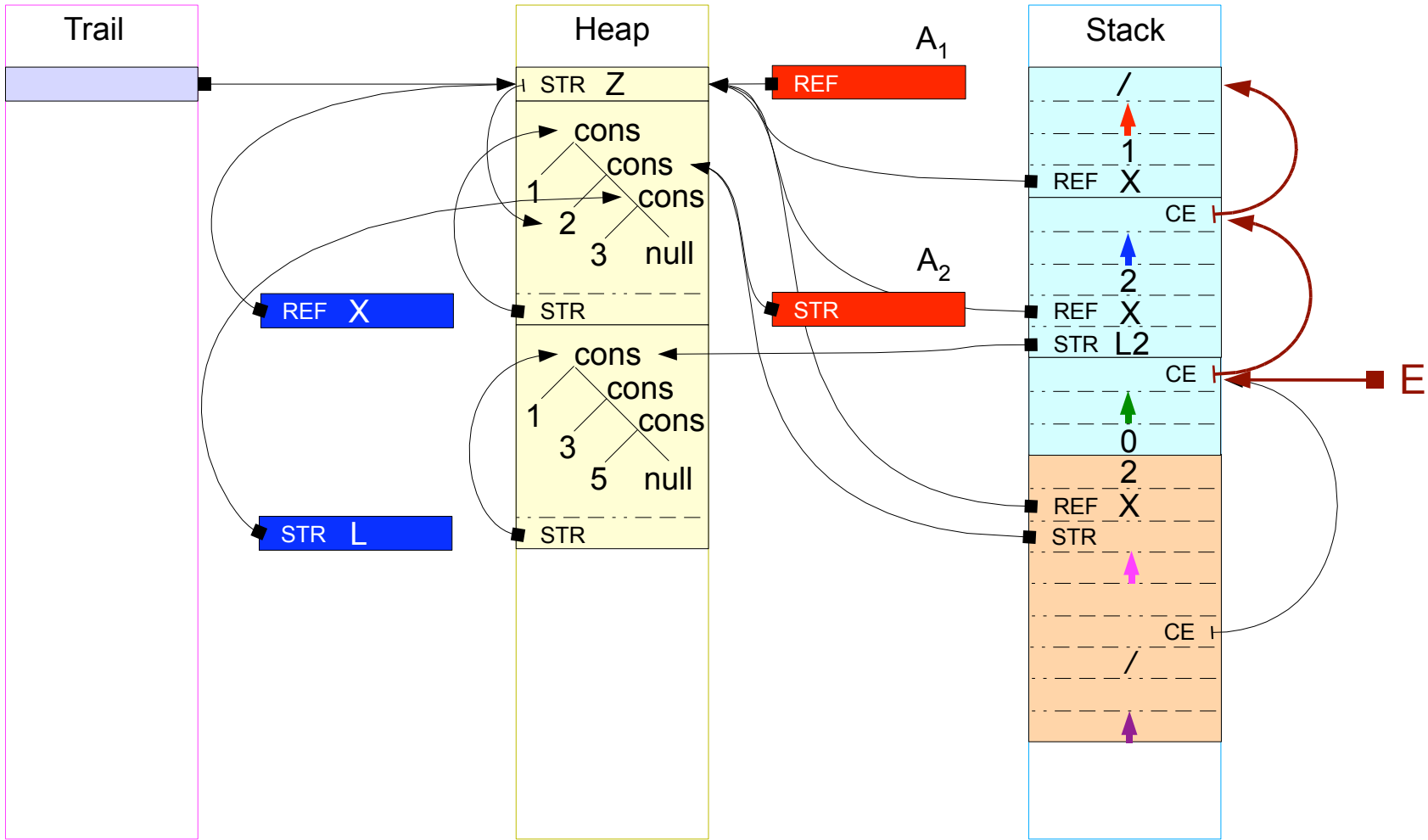
Trail

Heap

STR Z

cons
cons
cons
1
2
3 → null

REF X

STR

cons
cons
cons
1
3
5 null

STR L

STR

A_1

REF

A_2

STR

Stack

/
1
REF X
CE
2
REF X
STR L2
CE
0
CE
0
2
REF X
STR
CE
/

E

```
belong( X, cons(X,L) ).
belong( X, cons(Y,L) ) :- belong( X, L ).

shared( X, L1, L2 ) :- belong( X, L1 ), belong( X, L2 ).
chosen( X, L1, L2 ) :- shared( X, L1, L2 ), p( X ).

p( Y ) :- q( Y ).

q( 3 ).

?- chosen( Z, cons(1,cons(2,cons(3,null))),
           cons(1,cons(3,cons(5,null))) ).
```

```
belong( X, cons(X,L) ).
belong( X, cons(Y,L) ) :- belong( X, L ).

shared( X, L1, L2 ) :- belong( X, L1 ), belong( X, L2 ).
chosen( X, L1, L2 ) :- shared( X, L1, L2 ), p( X ).

p( Y ) :- q( Y ).

q( 3 ).

?- chosen( Z, cons(1,cons(2,cons(3,null))),
           cons(1,cons(3,cons(5,null))) ).
```

```
?- chosen( Z, [1,2,3], [1,3,5] ).
            |
?- shared( Z, [1,2,3], [1,3,5] ), p( Z ).
            |
?- belong( Z, [1,2,3] ), belong( Z, [1,3,5] ), p( Z ).
        {Z=1}
?- belong( 1, [1,3,5] ), p( 1 ).

?- p( 1 ).
      |
?- q( 1 ).
      |
    fail
```

```
?- belong( Z, [2,3] ), belong( Z, [1,3,5] ), p( Z ).
            {Z=2}
?- belong( 2, [1,3,5] ), p( 2 ).
              .
              .
              .
        fail
```

```
?- belong( 1, [3,5] ), p( 1 ).
          .
          .
          .
        fail
```

```
?- belong( Z, [3] ), belong( Z, [1,3,5] ), p( Z ).
          {Z=3}
?- belong( 3, [1,3,5] ), p( 3 ).
        |
?- belong( 3, [3,5] ), p( 3 ).
        |
?-  p( 3 ).
        |
?- q( 3 ).
        |
    succeed
```

```
?- belong( Z, [] ), ...
          ...
        fail
```

```
?- chosen( Z, [1,2,3], [1,3,5] ).
         |
?- shared( Z, [1,2,3], [1,3,5] ), p( Z ).
         |
?- belong( Z, [1,2,3] ), belong( Z, [1,3,5] ), p( Z ).
      {Z=1}
?- belong( 1, [1,3,5] ), p( 1 ).

                                    ?- belong( Z, [2,3] ), belong( Z, [1,3,5] ), p( Z ).
?- p( 1 ).                                        {Z=2}
      |                             ?- belong( 2, [1,3,5] ), p( 2 ).
?- q( 1 ).                                    .
      |                                       .
    fail                                      .
                                            fail

      ?- belong( 1, [3,5] ), p( 1 ).
                  .
                  .                     ?- belong( Z, [3] ), belong( Z, [1,3,5] ), p( Z ).
                  .                                 {Z=3}
                fail                    ?- belong( 3, [1,3,5] ), p( 3 ).
                                                    |
                                        ?- belong( 3, [3,5] ), p( 3 ).
                                                    |
                                        ?- p( 3 ).
                                                    |                 ?- belong( Z, [] ), ...
                                        ?- q( 3 ).                            ...
                                                    |
                                            succeed                         fail
```