```
(herald
 "Needham-Schroeder Protocol"
 (limit 200)  ;; Max # of skeletons to analyze
 (bound 10)   ;; Max # of strands in a skeleton
 ;; (algebra diffie-hellman)
 ;; include when using DH in defprotocol.
)
```

```
(defprotocol ns basic
  (defrole init
     (vars (a b name) (na nb text))
     (trace
       (send (enc a na (pubk b)))
       (recv (enc na nb (pubk a)))
       (send (enc nb (pubk b))))
     (uniq-orig na))
  (defrole resp … )
  (defrule high-trust-init
     (forall ((b name) (z strd))
      (implies
       (and
         (fact (high-trust-init))
         (p "init" z 0)
         (p "init" "b" z b))
       (non (privk b)))))
…)
```

```
(defskeleton ns basic
  (vars (alice bob name) (na nb text))
  (defstrand resp 3 (a alice) (b bob)
                  (na na) (nb nb))
  (deflistener (hash alice bob na nb))
  (non-orig (privk alice) (privk bob))
  (facts (high-trust-init))
)
```

Declarations for **Atoms** Only

```
(non-orig x)     ;; x secret and not carried
(uniq-orig x)    ;; x fresh at point first carried
(uniq-gen x)     ;; x fresh at point first used
(pen-non-orig x);; x secret
```

Other Declarations

```
(neq (x y))      ;; x != y
(eq (x y))       ;; x == y
(fn-of (function (x y))) ;; function(y) = x
```

Basic Cryptoalgebra

types: {**text**, **data**, **name**, **skey**, **akey**, **tag**} < mesg

functions:

```
    pubk:      name          -> akey
    privk:     name          -> akey
    invk:      akey          -> akey
    ltk:       name X name -> skey
    cat:       mesg X mesg -> mesg
    enc:       mesg X mesg -> mesg
    hash:      mesg          -> mesg
```

Cannot use a variable of sort mesg as the key in an encryption

Variables of sort mesg must be acquired (received before sent)

Types in **boldface** are **atom** types

Diffie-Hellman Cryptoalgebra

additional types: **rndx** < expt < mesg, base < mesg

additional functions:

```
    gen:          (none)     -> base
    exp:       base X expt -> base
    one:          (none)     -> expt
    rec:       expt          -> expt
    mul:       expt X expt -> expt
    bltk:      name X name -> skey
```

Variables of sort expt must be acquired (received before sent).

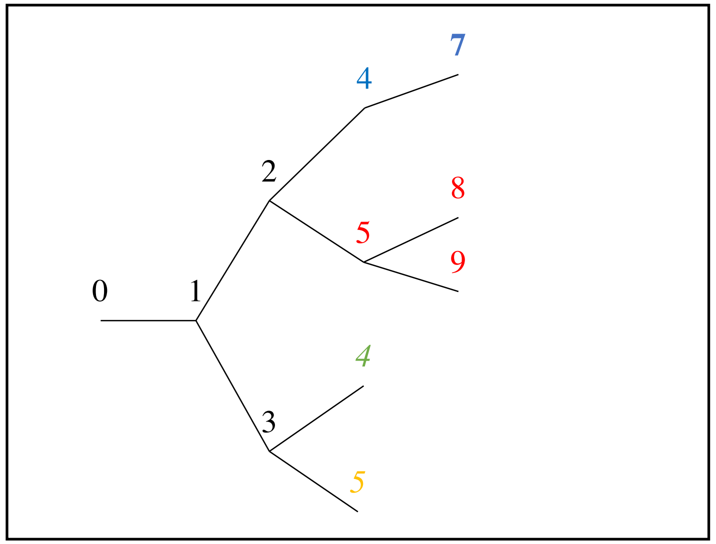
Rule / Goal atomic formulae

```
(p "role" z 2)         ;; instance/height
(p "role" "v" z v)     ;; instance parameter value
(non a)                ;; a is declared non-orig
(pnon a)               ;; a is declared pen-non-orig
(uniq a)               ;; a is declared uniq-orig
(uniq-at a z 2)        ;; a uniq-orig at node (z, 2)
(= v1 v2)              ;; equality
(prec z0 2 z1 3)       ;; (z0,2) precedes (z1, 3)
(leads-to z0 2 z1 3)   ;; (z0,2) leads to (z1, 3)
(fact pred params)     ;; User-defined facts
```

Rule / Goal grammar

<u>SENTENCE</u>	← (forall (GVDECL*) IMPLICATION)
GVDECL	← (ID+ SORT) | (ID+ strd)
IMPLICATION	← (implies CONJUNCTION CONCLUSION)
CONJUNCTION	← ATOMIC | (and ATOMIC*)
CONCLUSION	← (false) | EXISTENTIAL | (or EXISTENTIAL*)
EXISTENTIAL	← CONJUNCTION | (exists (GVDECL*) CONJUNCTION)

Interpreting Output



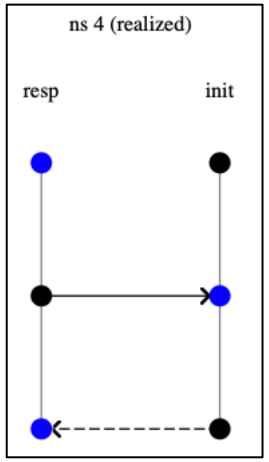
Key:

- Skeleton (partial execution)
- Realized Skeleton (full execution)
- Dead Skeleton (impossible partial execution)
- **Shape** (minimal full execution)
- *Seen Child* (links to elsewhere in the tree with live children)
- *Dead Seen Child* (links to elsewhere without live children)

●: send event
●: Unrealized recv event
●: Realized recv event
●: Unrealized obsv or tran event
●: init event or realized obsv or tran

→: Ordering with equal messages
⇢: Ordering with unequal messages
—: Strand ordering

Hover over a node to see its message
Hover over a role name to see the bindings
  of local variables



ns 4 (realized)
resp
init