Formal proofs related to incremental Merkle trees

Mizuhito Ogawa

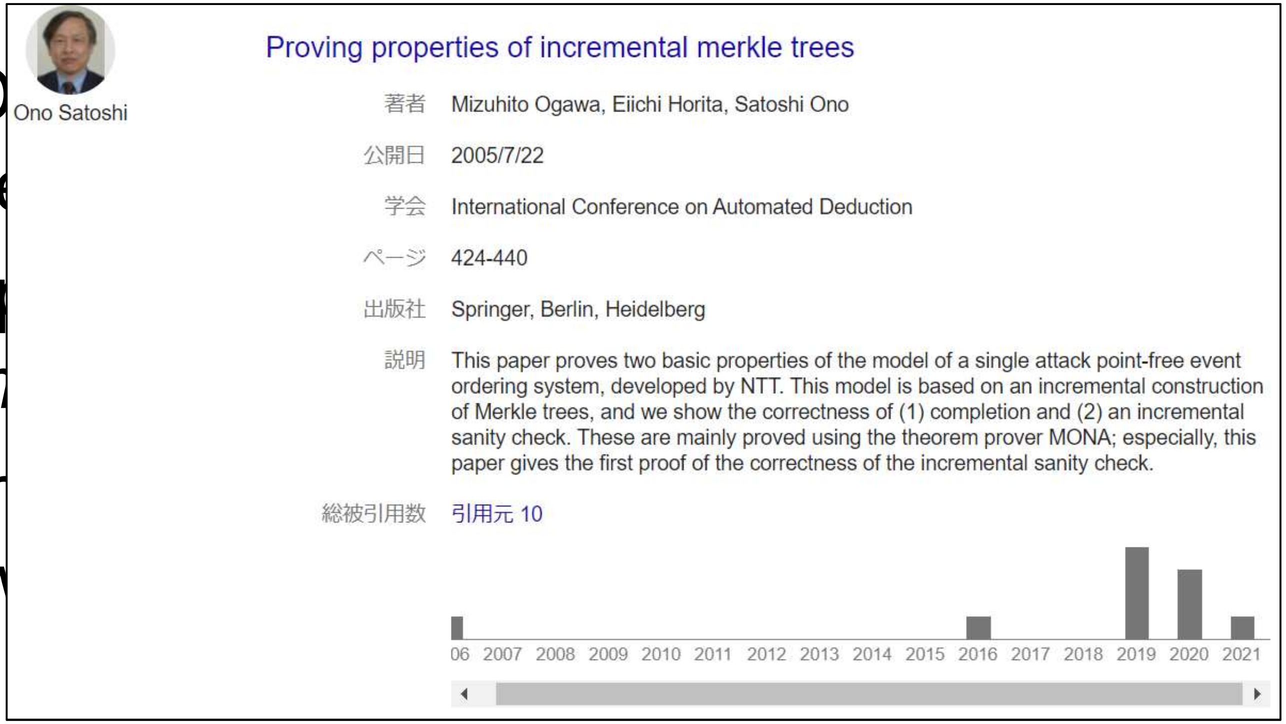
www.jaist.ac.jp/~mizuhito

TPP2021 北見工大

(originally in CADE 2005,
"*Proving properties of incremental Merkle trees*")

What we have shown in CADE 2005?

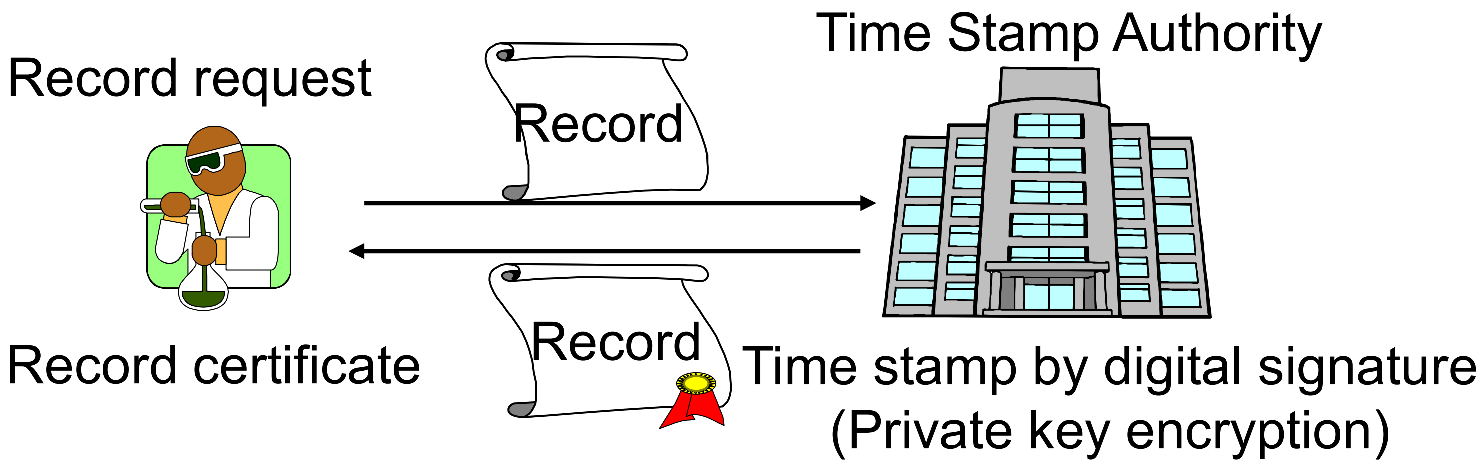
- M Ogawa, S O
  incremental Me
  - ✓NTT develop
    like *block ch* .
  - ✓Proved “Sar
    by MONA (w



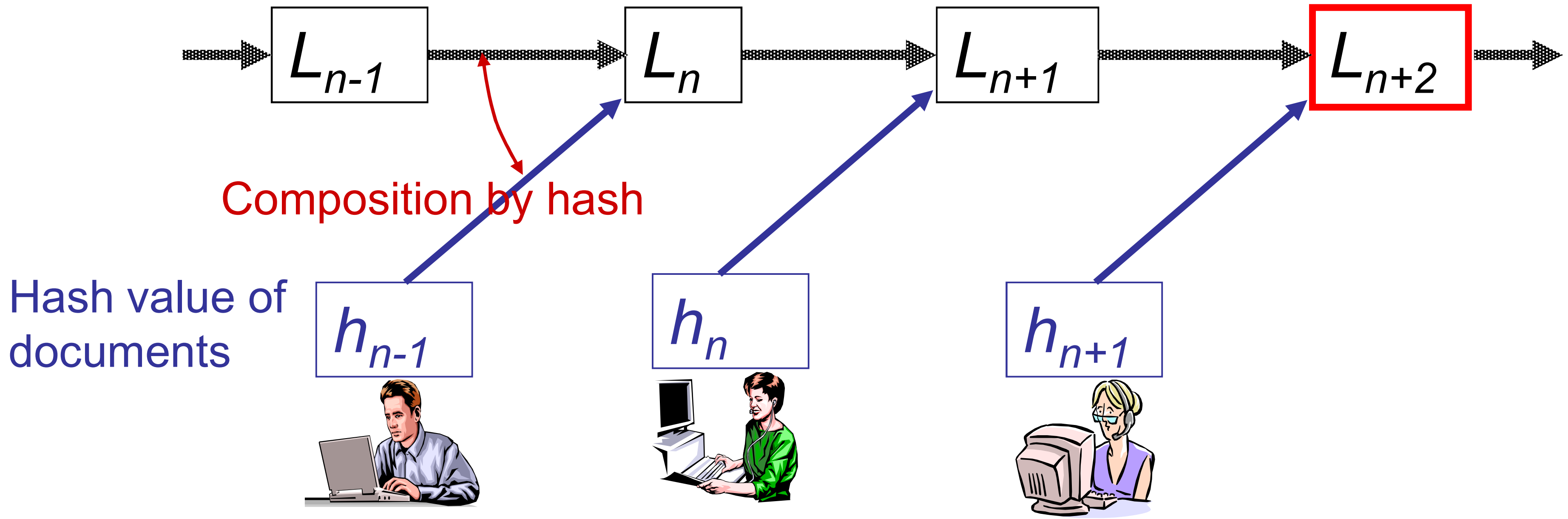
- Considering recent citation in formal proofs, (ESORICS 2016, ITP 2019/2020, ArXiv 2021, …)
  - ✓Complete formal proof in *Isabelle sledgehammer?*
  - ✓How to formalize properties of *Hash function*?
  - ✓How to formalize consensus? (e.g., majority)

Two methods for event-ordering certificate

Time stamp by digital signature (rfc-3161)

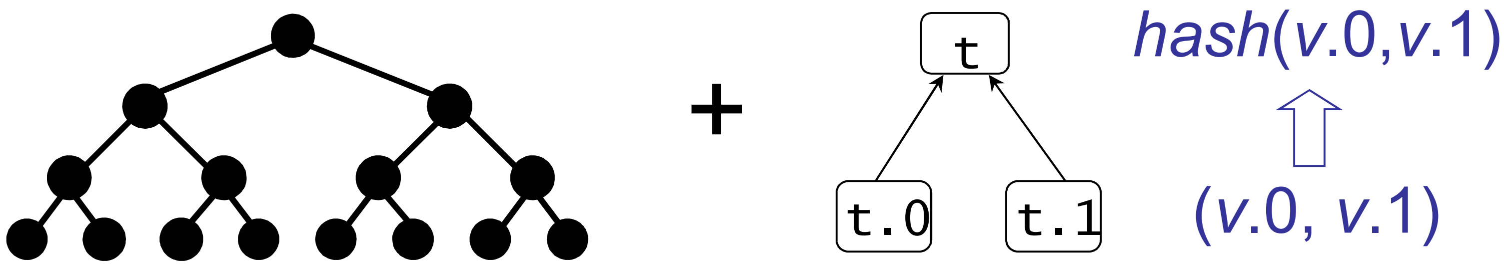


Linking and publication by hash function (ISO18014-3)



Merkle tree (Merkle 1979)

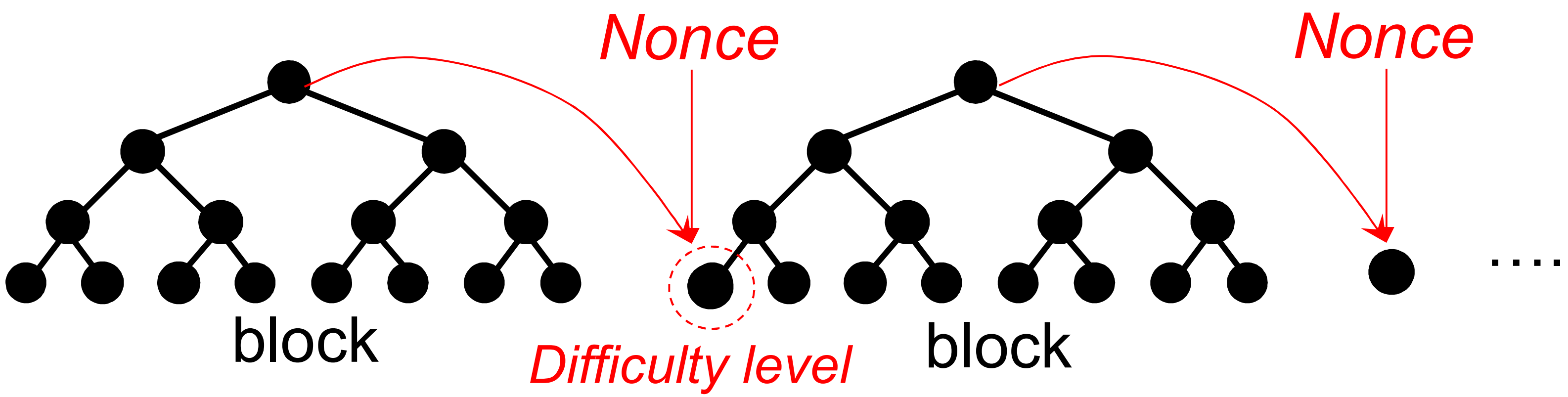
- *Merkle tree = Binary tree + hash function*
- Each node has its hash value, computed from a pair of hash values of its children.



We assume collision-resistance, one-way hash function.

Block chain = IML + *finding nonce* + DLT

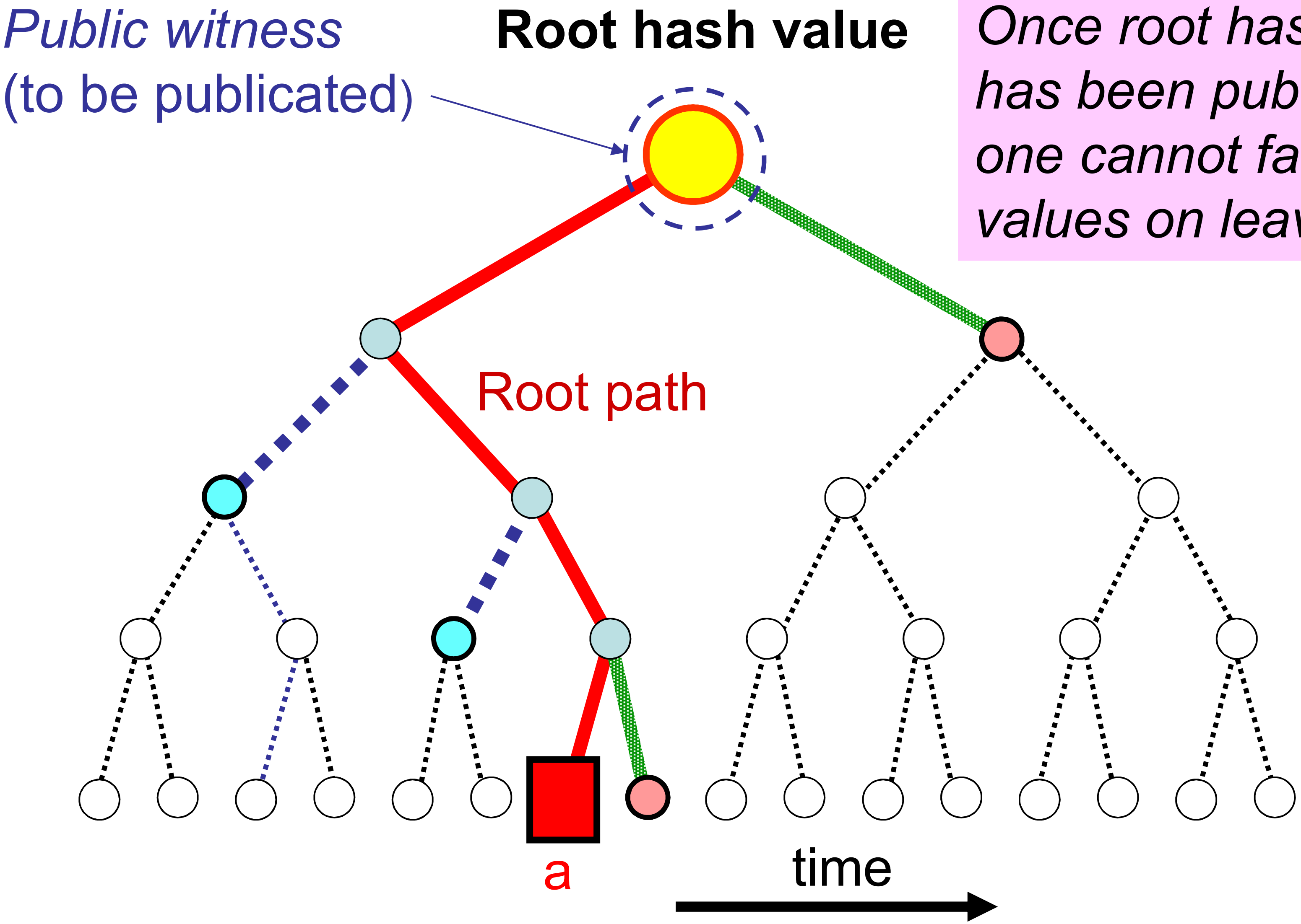
- Incremental Merkle Tree (IML)



- Finding *nonce* (mining)
  - ✓Nonce such that its hash value holds *difficulty level.*

- DLT = distributed ledger technology
  - ✓Majority decision.
  - ✓Longer block chain is “*more*” valid.

Basic idea : Merkle tree (1)



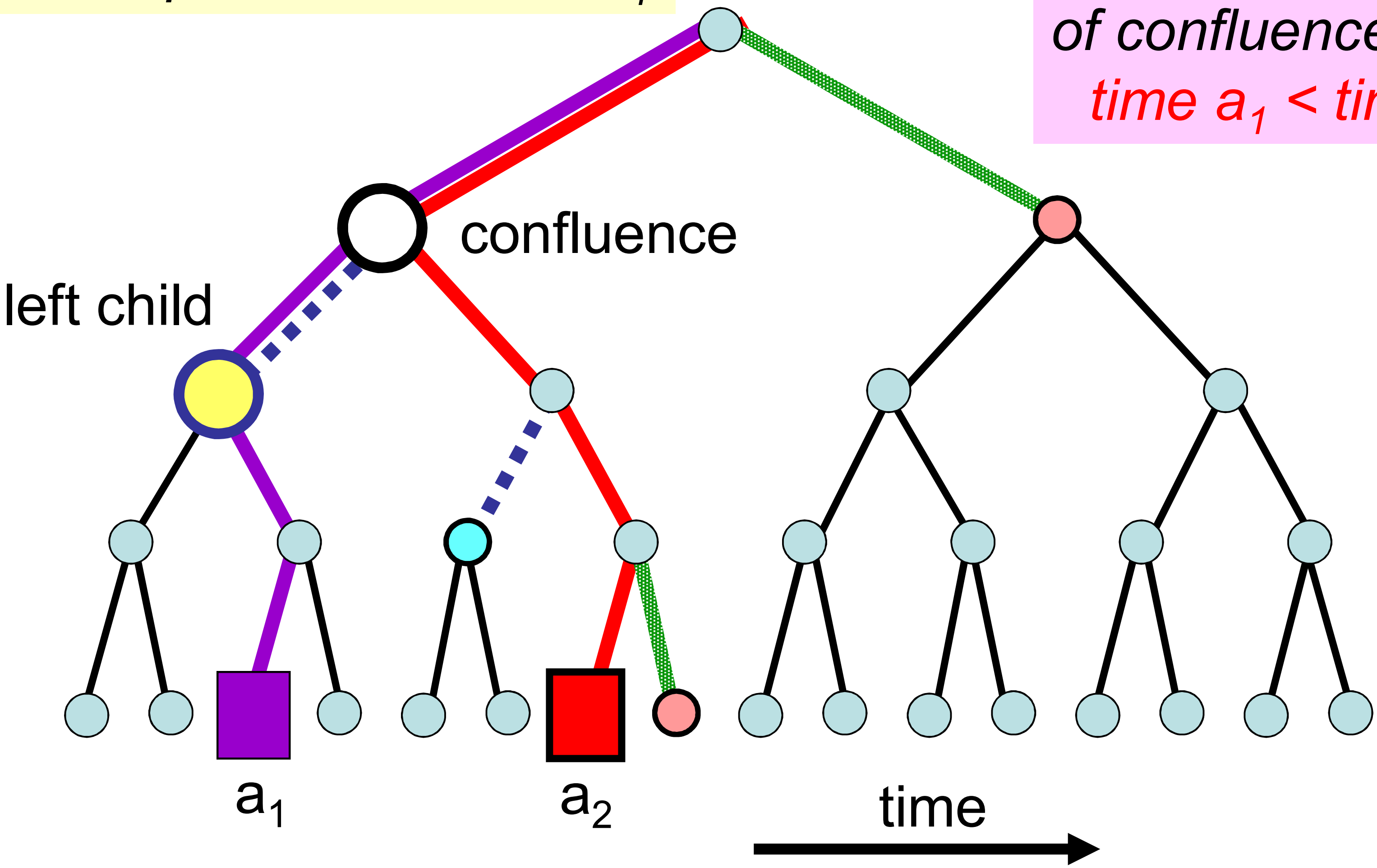
Once root hash value has been publicated, one cannot falsify hash values on leaves.

Basic idea: Merkle tree (2)

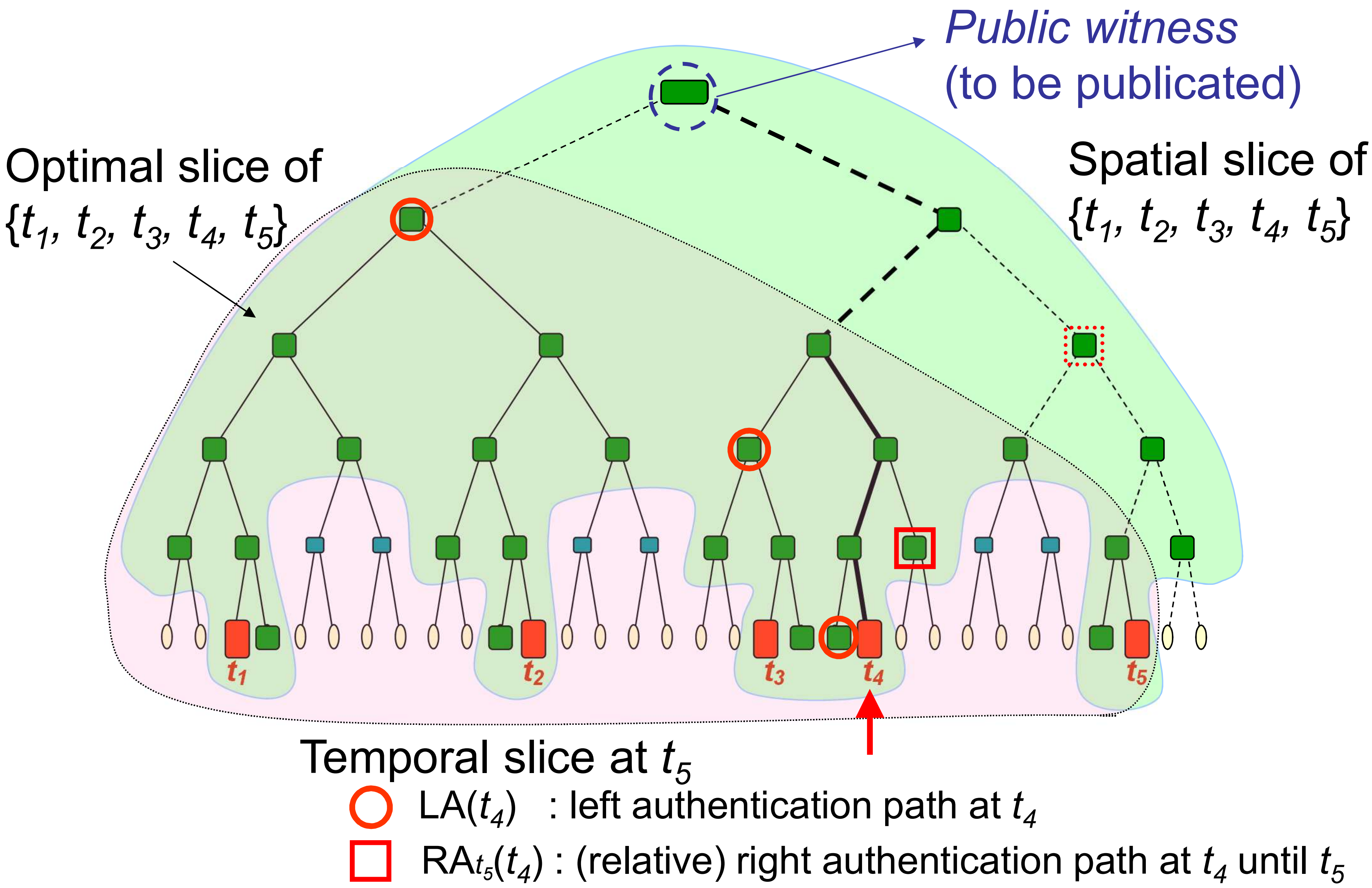
The hash value of the left child depends on that of a_1.

If the root path of a_1 contains the left child of confluence with a_2

time a_1 < time a_2



Incremental Merkle trees construction for registration requests at t_1, t_2, t_3, t_4, t_5



Protocol for event registrations

- Assume a user registers at $t_1, t_2, \ldots, t_n$ and receives:
  - ✓ $(\phi, LA(t_1) \cup \{t_1\})$ at t_1.
  - ✓ $(RA_{t_i}(t_{i-1}), LA(t_i) \cup \{t_i\})$ at t_i with $0 < i \leqq n$.

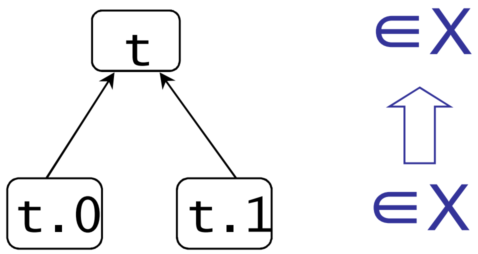
  where $LA(t)$ is the set of left authentications, and
  $RA(t)$ is the set of right authentications.

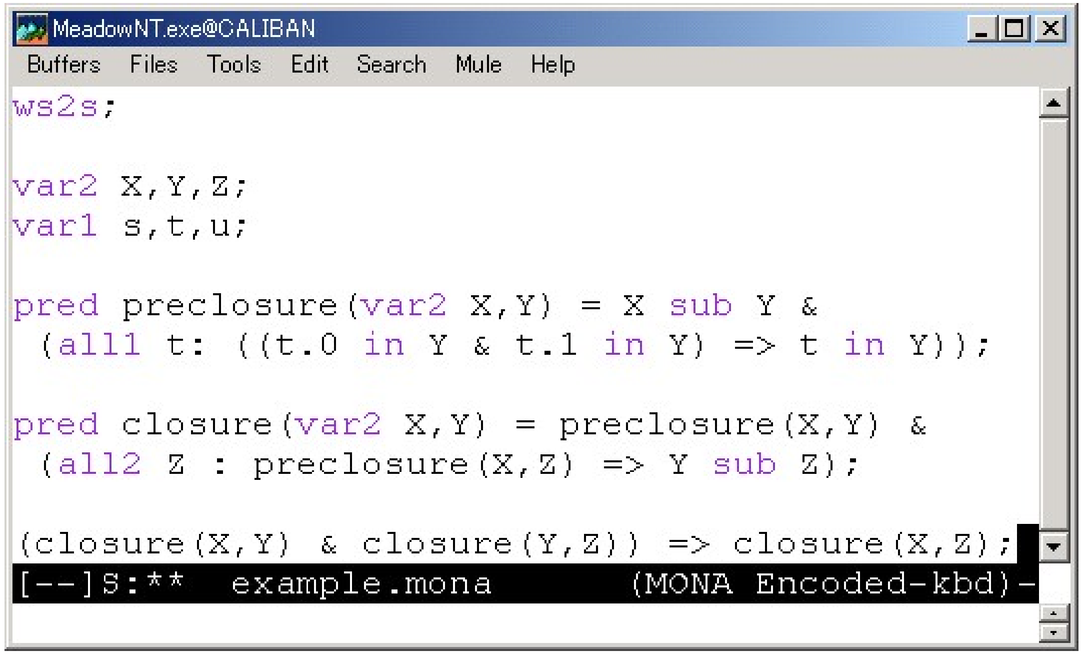
Described in WS2S
(incomparable(A) & opt_slice(A,X) & LSRclosure_union(A,Y)) => X = Y;

- **Th.** OptimalSlice($\{t_1, t_2, \ldots, t_n\}$)
  $= (\bigcup_{1 \leqq i < n} Cls(LSR_{t_{i+1}}(t_i))) \cup Cls(LS(t_n))$

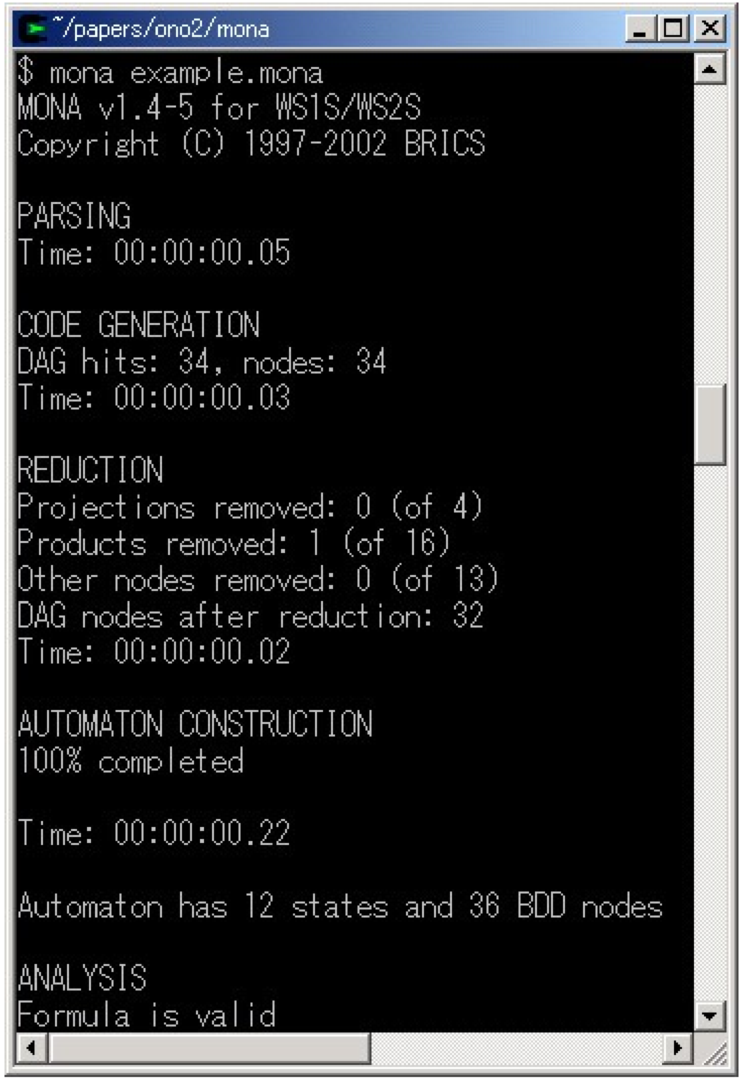
  where $LS(t_i) = LA(t_i) \cup \{t_i\}$, $LSR_{t_{i+1}}(t_i) = LS(t_i) \cup RA_{t_{i+1}}(t_i)$,
  Closure $Cls(X)$ is the minimum set with $X \subseteq Cls(X)$ and
  - ✓ t.0 (left child), t.1 (right child) $\in Cls(X) \Rightarrow t \in Cls(X)$

Recall: MONA example on closure



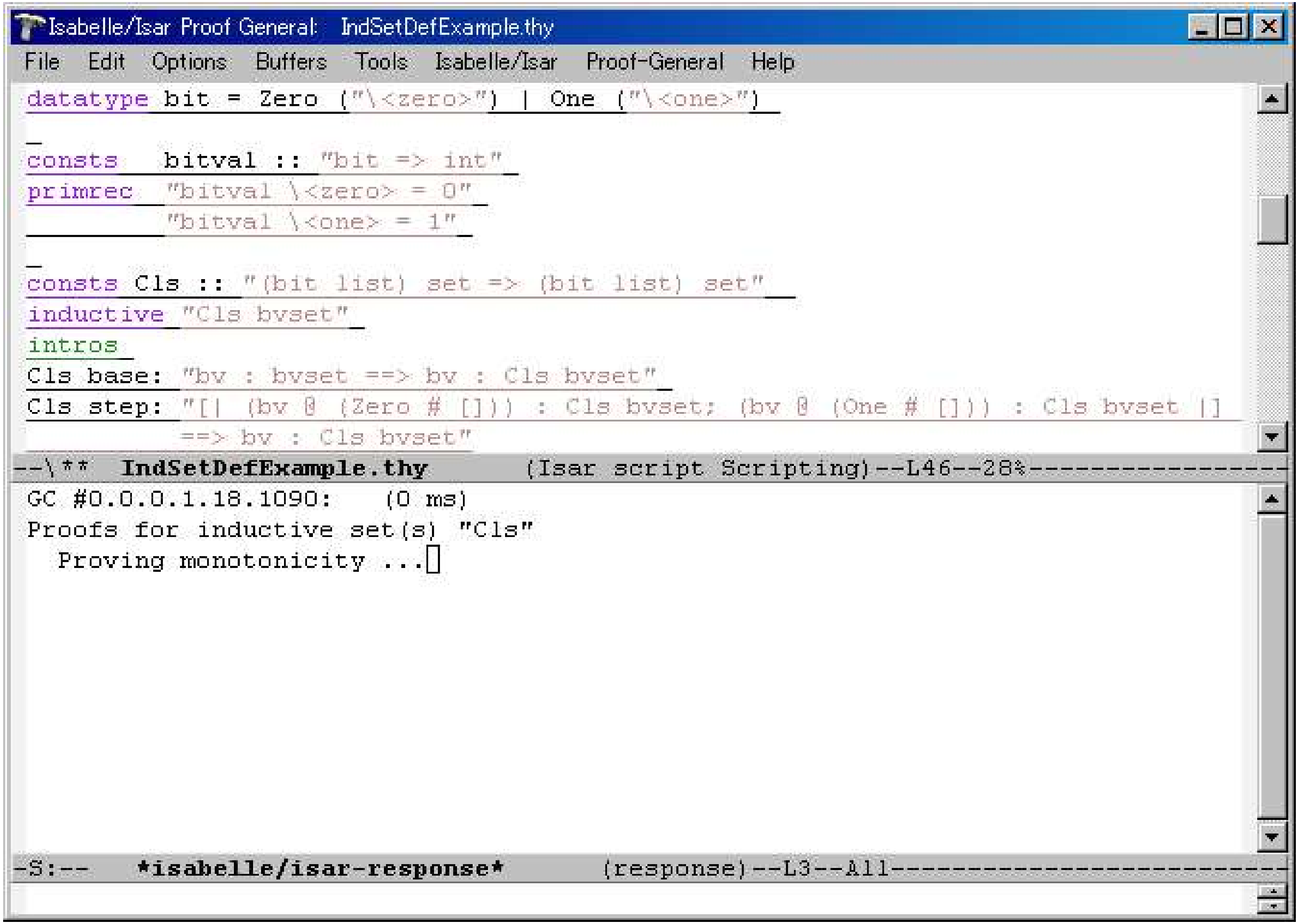



```
ws2s;

var2 X,Y,Z;
var1 s,t,u;

pred preclosure(var2 X,Y) = X sub Y &
 (all1 t: ((t.0 in Y & t.1 in Y) => t in Y));

pred closure(var2 X,Y) = preclosure(X,Y) &
 (all2 Z : preclosure(X,Z) => Y sub Z);

(closure(X,Y) & closure(Y,Z)) => closure(X,Z);
```

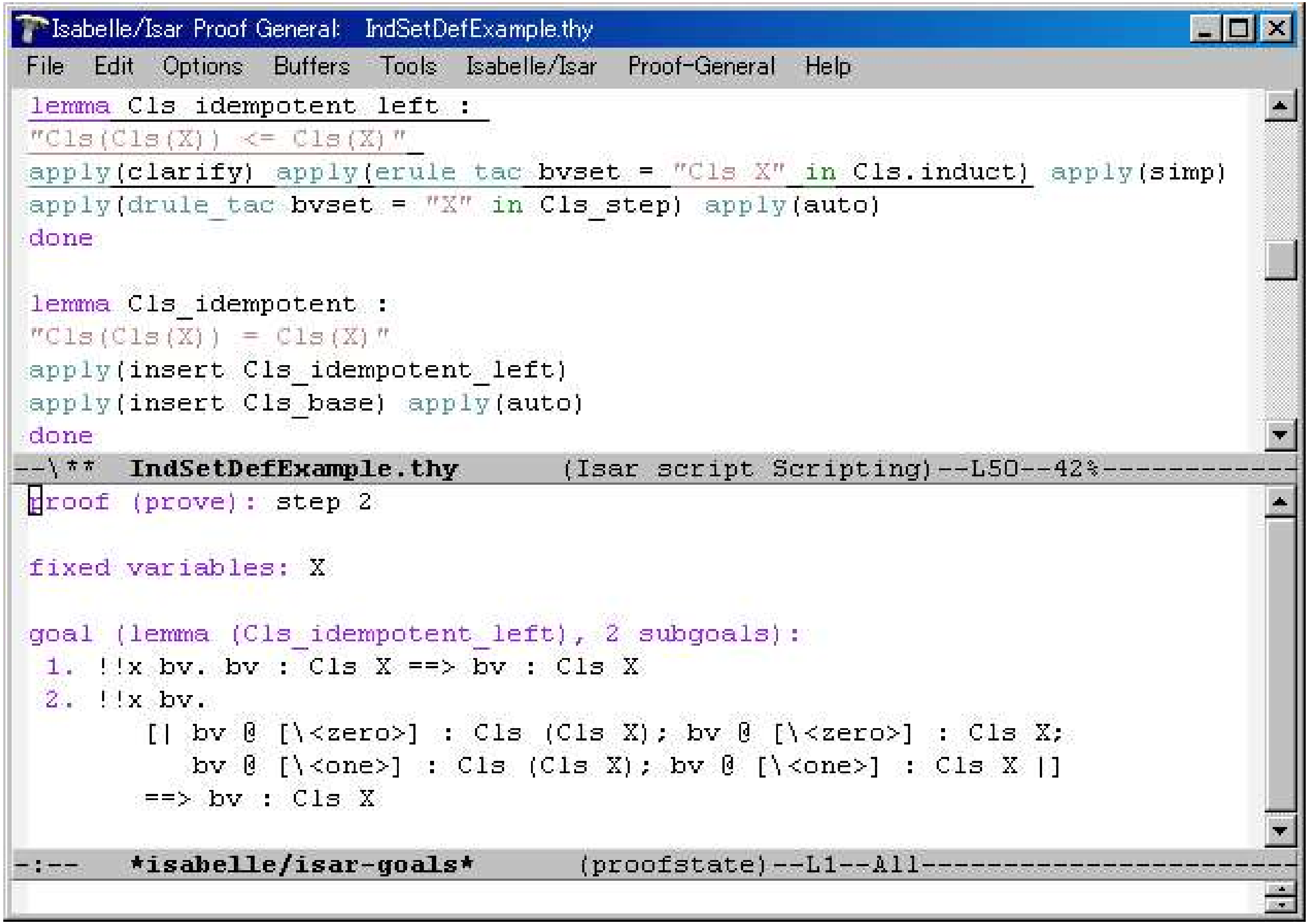



```
$ mona example.mona
MONA v1.4-5 for WS1S/WS2S
Copyright (C) 1997-2002 BRICS

PARSING
Time: 00:00:00.05

CODE GENERATION
DAG hits: 34, nodes: 34
Time: 00:00:00.03

REDUCTION
Projections removed: 0 (of 4)
Products removed: 1 (of 16)
Other nodes removed: 0 (of 13)
DAG nodes after reduction: 32
Time: 00:00:00.02

AUTOMATON CONSTRUCTION
100% completed

Time: 00:00:00.22

Automaton has 12 states and 36 BDD nodes

ANALYSIS
Formula is valid
```

Definition of closure



Example lemma: closure is idempotent



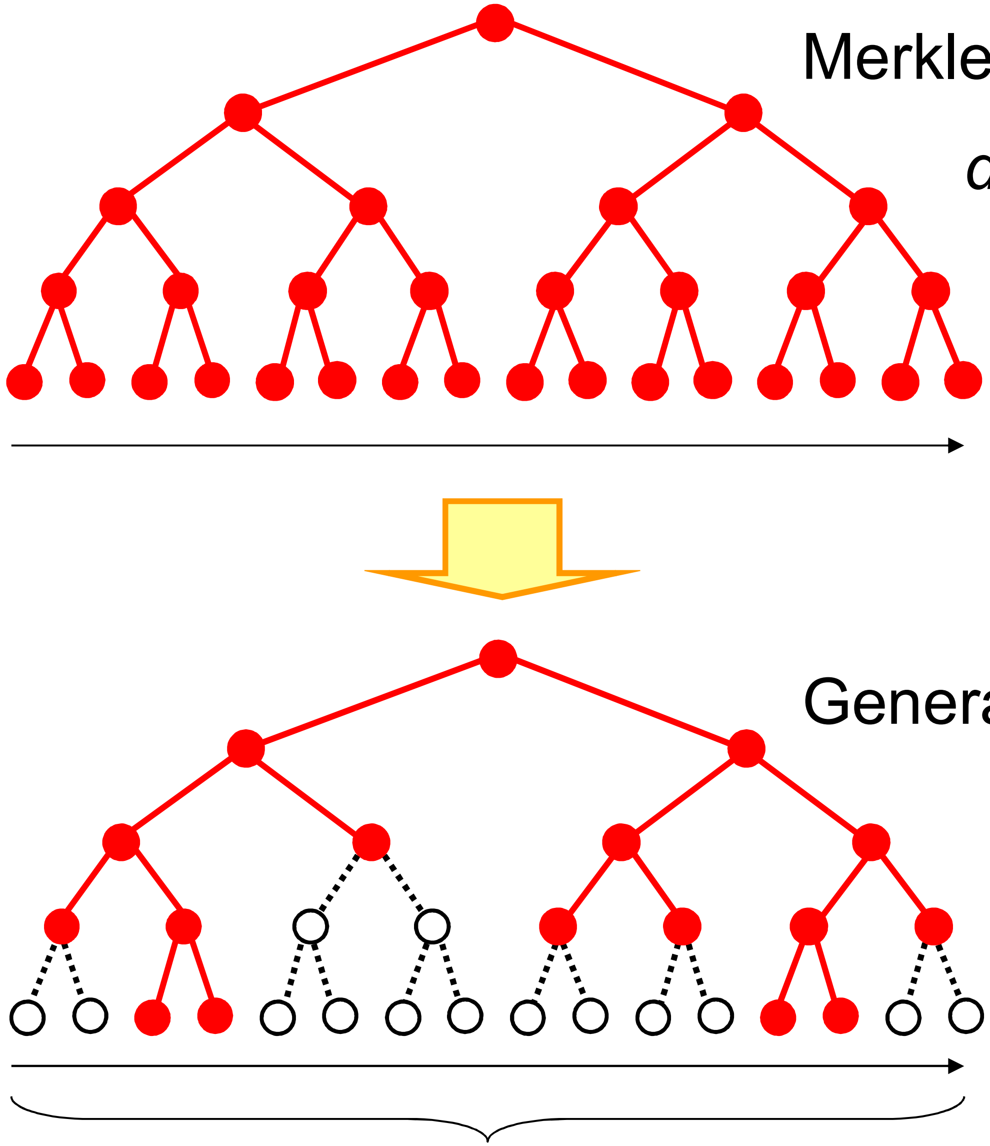
MONA Trick 1 : Generalized Merkle tree

- MONA cannot describe that :
  "*a binary tree has the same depth*"
  (i.e., *each root path has the same length*)

- We have been implicitly assuming that :
  "*a Merkle tree has the same depth*"

- We relax "*the same depth*" to just "*don't care depth*".

Generalized Merkle tree example



Merkle tree

depth is the same

Generalized Merkle tree

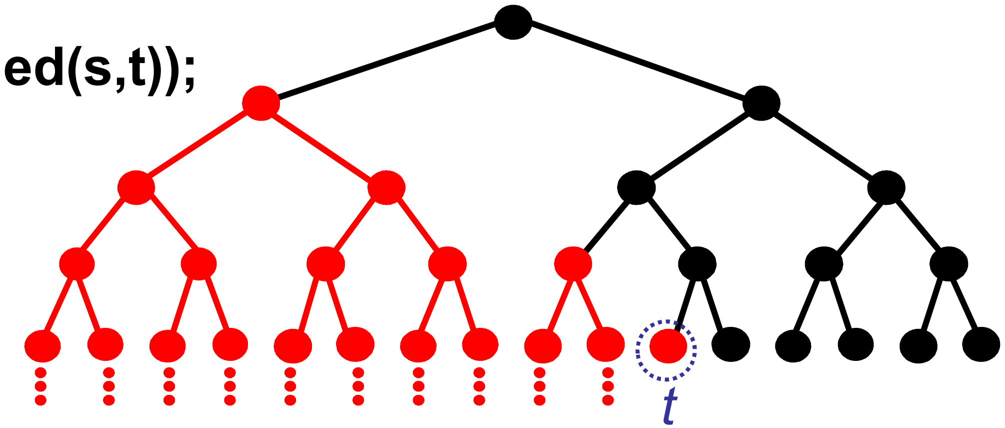
t.0 ∈ T <=> *t.1*∈ T

incomparable, left-to-right

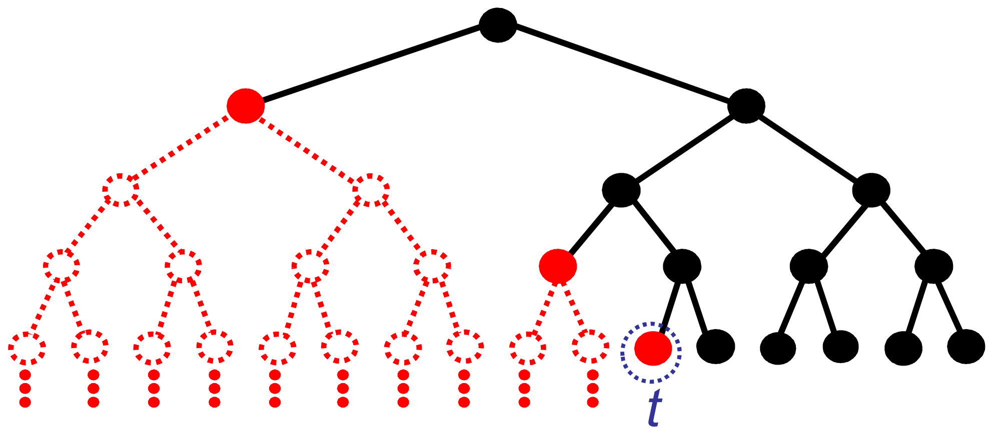
MONA Trick 2: *Temporal slice* in WS2S

- First attempt : "Subtrees can grow infinitely." (S2S)

all1 s: (s in X <=> defined(s,t));

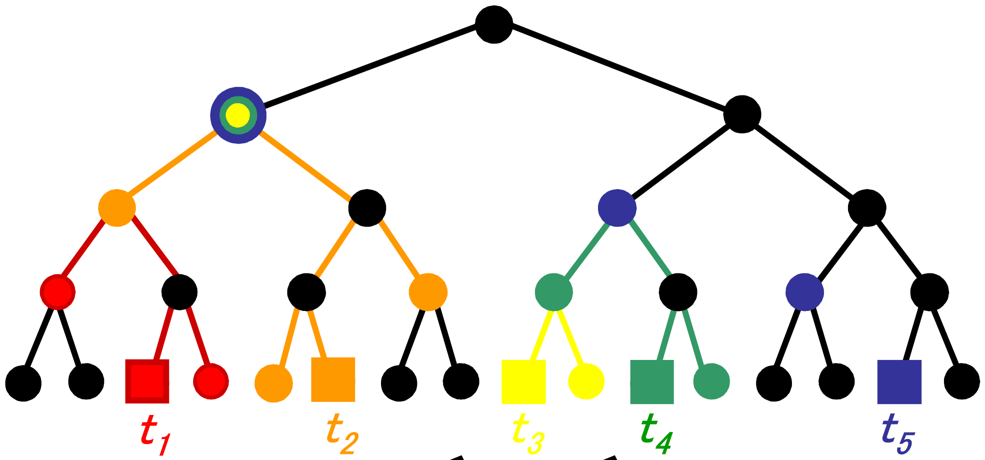


- Second attempt : "*temporal slice as its roots*."



Sanity Check

- Hash values are computed from different LSR$t_{i+1}(t_i)$'s (i.e., different users compute hash values individually).
  - ✓ If multiple computations at each node coincide (i.e., *consistent*), *it suggests no internal-failures*.



- **Key Lemma**. Let $i+1 \leqq k \leqq j$. Then, $\mathrm{Cls}(\mathrm{LSR}t_{i+1}(t_i)) \cap \mathrm{Cls}(\mathrm{LSR}t_{j+1}(t_j)) \subseteq \mathrm{Cls}(\mathrm{LS}(t_k))$

Described in WS2S

(lefter(s,t) & (t = u | lefter(t,u)) & (u = v | lefter(u,v)) & (v = w | lefter(v,w))
& LSRclosure(s,t,X) & LSclosure(u,Y) & LSRclosure(v,w,Z))
=> X inter Z sub Y;

Consistency

- Let (U_i, α_i) such that
  ✓ U_i : a set of incomparable nodes,
  ✓ α_i : labeling function on U_i
  (extended $\alpha_i(t) = \mathit{hash}(\alpha_i(t.0), \alpha_i(t.1))$ on $\mathrm{Cls}(U_i)$)

Beyond WS2S (MONA)

- **Def.** $\{(U_i, \alpha_i)\}$ is *weakly consistent* if
  $\alpha_i(t) = \alpha_j(t)$ for each $t \in \mathrm{Cls}(U_i) \cap \mathrm{Cls}(U_j)$

- **Def.** $\{(U_i, \alpha_i)\}$ is *consistent* if α holds

$$\alpha(t) = \begin{cases} \alpha_i(t) & \text{when } t \in U_i \\ \mathit{hash}(\alpha(t.0), \alpha(t.1)) & \text{when } t \notin \mathrm{leaves}(\cup U_i) \end{cases}$$

Incremental Sanity Check

- **Lemma 1.** $\{(\mathrm{LSR}t_{i+1}(t_i), \alpha_i) \mid 1 \leqq i < n\} \cup \{(\mathrm{LS}(t_n), \alpha_n)\}$ is *weakly consistent,* if $\{ (\mathrm{LSR}t_{i+1}(t_i), \alpha_i), (\mathrm{LS}(t_{i+1}), \alpha_{i+1}) \}$ is *weakly consistent* for each $1 \leqq i < n$.

- **Lemma 2.** $\{(\mathrm{LSR}t_{i+1}(t_i),\alpha_i) \mid 1 \leqq i<n\} \cup \{(\mathrm{LS}(t_n),\alpha_n)\}$ is *consistent*, if it is *weakly consistent*.

  Proof. Let X be the optimal slice of $\{t_1,t_2, \ldots, t_n\}$.

  By induction on $|X \cap T_t|$ for $T_t = \{s \mid t \leqq s\}$.

  Beyond WS2S (MONA)

- **Cor.** $\{(\mathrm{LSR}t_{i+1}(t_i),\alpha_i) \mid 1 \leqq i<n\} \cup \{(\mathrm{LS}(t_n),\alpha_n)\}$ is *consistent,* if $\{ (\mathrm{LSR}t_{i+1}(t_i), \alpha_i), (\mathrm{LS}(t_{i+1}), \alpha_{i+1}) \}$ *is weakly consistent for each i with* $1 \leqq i < n$.

  Each user correctly computes

Discussion

- Early formal proof example on incremental Merkle tree.
  - ✓Mainly for "*Sanity check*"
- Isabelle sledhammer
  - ✓FOL provers, SMT solvers, but not for MONA.
  - ✓How to generate Isabelle proof for MONA validity?
- More properties
  - ✓DLT neglected.
  - ✓*Almost everywhere, consensus* (⇒ Belief?)