Remote Belief: Preserving Volition for Loosely Coupled Processes

Nuh Aydin (Kenyon College)
Paul Sivilotti (Ohio State University)

http://www.cis.ohio-state.edu/~paolo

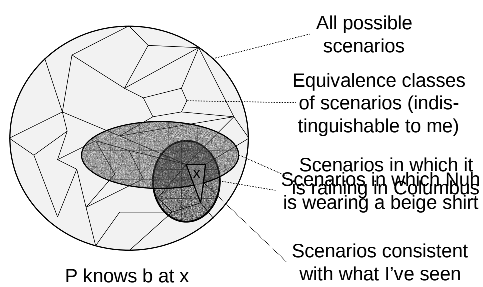
Conclusions

- Knowledge too strict for loosely coupled systems
  - Entails an undesirable loss of volition
- Define "belief" in distributed systems
  - Computation isomorphism
  - Use limits to handle unbounded computations
- Belief transfer is more permissive
  - Belief can <u>increase</u>/decrease with send
  - Belief can increase/<u>decrease</u> with receive
- Simple theorems for manipulating belief
  - But naive interpretations as probability are dangerous
- Examples
  - Directories for distributed components
  - Asynchronous leases

Knowledge: Background

- What does it mean to "know"?
  - Philosophy, economics, game theory, …
  - CS: AI, cog. science, distributed systems, …
- Common thread:
  - indistinguishable worlds relative to a principal
- "I know Nuh is wearing a beige shirt"
  - Many *possible* states of the world right now
    - Could be sunny/raining in Columbus, OH
    - Could be sunny/raining in Pasadena, CA
  - But in *all* of them, Nuh's shirt is beige

Indistinguishable Worlds



P knows b at x
$\equiv (\forall y : y \in [P]_x : b \text{ at } y)$

Distributed Systems

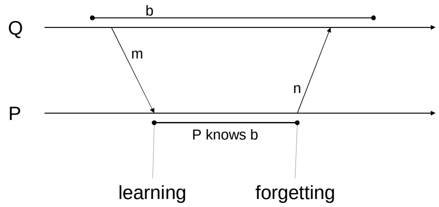
- Computation consists of 3 kinds of actions:
  - Internal actions, sends, and receives
- Indistinguishable worlds
  - P only sees actions *local to P*
  - $x = \langle e_P e_Q s_P r_Q e_P \rangle$ $y = \langle e_P s_P e_P r_Q \rangle$ $z = \langle e_Q e_P s_P e_P \rangle$ $(e_P s_P e_P)$
- Interesting case: b is remote
  - From its observations, P can conclude b at Q
- Example: P knows Q has sent message m
  - P receives m from Q
- Example: P knows printer's paper tray is empty
  - P receives "out of paper" from printer

Knowledge Transfer

(For asynchronous systems and remote predicates)

- Knowledge is not gained or lost by internal actions
  - (P knows b at x) = (P knows b at $(x;e_P)$)
- Knowledge can not be gained by sends
  - (P knows b at x) $\Leftarrow$ (P knows b at $(x;s_P)$)
  - *Knowledge is gained only by receives*
  - P receives "paper tray is empty"
- Knowledge can not be lost by receives
  - (P knows b at x) $\Rightarrow$ (P knows b at $(x;r_P)$)
  - *Knowledge is lost only by sends*
  - P sends "paper can be added to tray"

Learning and Forgetting



Applications

- Two generals problem
  - P knows (Q knows (P knows (Q knows… b)))
  - Common knowledge can not be gained/lost
- Message complexity lower bounds
  - Detector knows computation has terminated
  - Requires (chain of) messages from all parties
- Distributed design by contract
  - Client must establish precondition before op
    - Assert: pre
    - Client knows pre

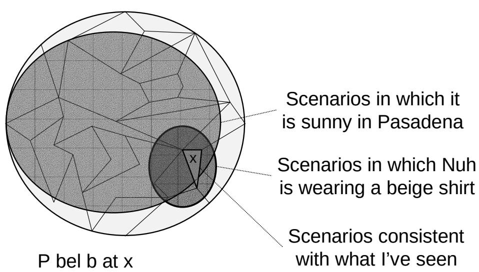
Loss of Volition

- Once P knows b, for some remote b (on Q)
  - It must be true in all (isomorphic) computations
  - Q can not unilaterally violate b!
- Example:
  - P knows printer paper is empty
  - Printer is not allowed to add paper, *without permission from P*
- If P knows something about Q, Q has *lost volition* over that part of its state
  - Even sending a message does not allow Q to change that part of its state!

Loosely Coupled Systems

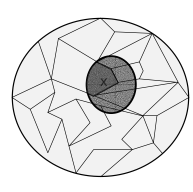
- Some state can (should) not be controlled
  - Failure state is outside printer's control
    - "printer is online"
  - Clients should be able to empty paper tray
    - "there is paper in the tray"
- Some processes may be unreliable
  - Q must wait for permission from P
  - This permission may never come!
    - Crash, unreliable connectivity, maliciousness
- Knowledge may be inappropriate for loosely coupled systems
  - Soft-state, heart beat algorithms

Belief: Preserving Volition



P bel b at x
≡ (for "most" y : y ∈ $[P]_x$: b at y)

Defining Belief



$[P]_x$: comp's isomorphic to x

$[P_b]_x$: comp's isomorphic to x, and for which b holds

P knows b at x ≡ ($[P_b]_x = [P]_x$)

P bel_α b at x ≡ ($|[P_b]_x| / |[P]_x| \geq \alpha$)

Problem: computations may be unbounded, so sets may be infinite (& quotient is indeterminate)

Defining Belief

- Solution: use limit
  - T_N = set of computations of length <= N
  - Sequence $a_N = |T_N \cap [P_b]_x| / |T_N \cap [P]_x|$
  - P bel_α b at x $\equiv (\lim_{N \to \infty} a_N \geq \alpha)$
- Another problem: limit may not exist
  - Solution: use lim inf (or lim sup)
  - No reason to prefer one over the other
    - Average
  - P bel_α b at x $\equiv$ ½(lim $\inf_{N \to \infty} a_N$ + lim $\sup_{N \to \infty} a_N$) $\geq \alpha$

Belief is not Knowledge

- P knows b at x $\Rightarrow$ P bel_1 b at x
  - But not vice versa!
  - b holds in the limit, as longer computations in $[P]_x$ are considered
  - b holds for *all* computations in $[P]_x$
- Example:
  - P and Q have only internal events
  - b = 2 or more internal events at Q
  - P bel_1 b at ε, but $\neg$(P knows b at ε)
- Exception: when computations are bounded

Working with Knowledge

P knows b $\Rightarrow$ b

(P knows b) $\wedge$ (b $\Rightarrow$ b') $\Rightarrow$ P knows b'

(P knows b) $\wedge$ (P knows b') $\Rightarrow$ P knows (b $\wedge$ b')

(P knows b) $\vee$ (P knows b') $\Rightarrow$ P knows (b $\vee$ b')

P knows (P knows b) = P knows b

P knows ($\neg$(P knows b)) = $\neg$(P knows b)

Working with Belief

P bel_α b $\Rightarrow$ P bel_β b, $\beta \leq \alpha$

P bel_0 b

P bel_α b $\wedge$ (b $\Rightarrow$ b') $\Rightarrow$ P bel_α b'

(P bel_α b) $\wedge$ (P bel_β b') $\Rightarrow$ P $bel_{max\{0,\alpha+\beta-1\}}$ (b $\wedge$ b')

(P bel_α b) $\wedge$ (P bel_β b') $\Rightarrow$ P $bel_{max\{\alpha,\beta\}}$ (b $\vee$ b')

(P bel_α b) $\vee$ (P bel_β b') $\Rightarrow$ P $bel_{min\{\alpha,\beta\}}$ (b $\vee$ b')

P bel_α b = P knows (P bel_α b)

P bel_α (Q knows b) $\Rightarrow$ P bel_α b

P bel_α b = P $bel_{1-\alpha}$ $\neg$b, for maximal α

Belief Transfer: Receives

- Belief can increase or *decrease* as a result of a receive
- Example
  - Q has coin, initially heads or tails
  - Q sends this value to P
  - Initially:
    - P $bel_{1/2}$ Q has heads
    - P $bel_{1/2}$ Q has tails
  - After P receives message "heads":
    - P bel_1 Q has heads
    - P bel_0 Q has tails

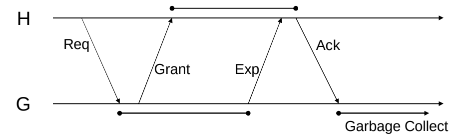
Belief Transfer: Sends

- Belief can decrease or *increase* as a result of a send
- Example
  - Q has a coin, initially in random state
  - Q receives message and sets coin accordingly
  - Initially:
    - P $bel_{1/2}$ Q has heads
    - P $bel_{1/2}$ Q has tails
  - After P sends message "set to heads"
    - P bel_1 Q has heads
    - P bel_0 Q has tails

Eg: Directories

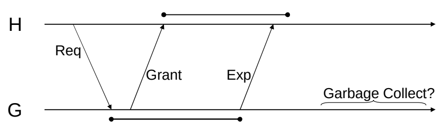
- RMI registry, CORBA Naming, UDDI
- Contains information about remote objects
  - IP address, string name, interface information
  - Initialized by receiving a message
- This entry does not reflect knowledge
  - Object would require directory's permission to change its attributes
- Instead, entry reflects *belief*
  - Directory believes remote objects have given properties, with appropriate threshold α

Eg: Asynchronous Leases



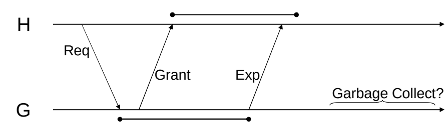
- Leases can be viewed as knowledge
  - After receiving "Grant", H knows ref is valid
  - After receiving "Ack", G knows ref is released
- Example: garbage collection
  - What if H crashes (and "Ack" never arrives)?

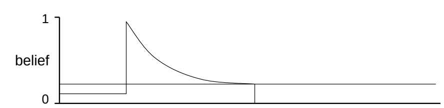
Eg: Asynchronous Leases



- Instead, leases reflect *belief*
  - After sending "Grant", G bel_1 H holds ref
  - As time elapses, this belief decays
    - Decay depends on likelihood of H crashing
  - When some threshold is reached, G sends "Exp"
    - After send, G bel_1 ref is released

Eg: Asynchronous Leases





Conclusions

- Knowledge too strict for loosely coupled systems
  - Entails an undesirable loss of volition
- Defined "belief" in distributed systems
  - Computation isomorphism
  - Limits to handle unbounded computations
- Belief transfer is more permissive
  - Belief can <u>increase</u>/decrease with send
  - Belief can increase/<u>decrease</u> with receive
- Simple theorems for manipulating belief
  - But naive interpretations as probability are dangerous
- Examples
  - Directories for distributed components
  - Asynchronous leases

Future Work

- Probabilities and measures
  - Computations can be weighted according to some probability model
  - Use more general measures
- Investigation of chained belief
  - $P_1\ bel_{\alpha 1}\ (P_2\ bel_{\alpha 2}\ (P_3\ bel_{\alpha 3} \ldots (P_n\ bel_{\alpha n}\ b)))$
- Investigation of common belief
  - Reformulations of consensus around belief

Acknowledgements

- Distributed Components research group at Ohio State
  - Scott Pike, Nigamanth Sridhar, Murat Demirbas, Hilary Stock, Chris Bohn
- Funding sources:
  - National Science Foundation (ITR)
  - SBC Ameritech
  - Lucent Technologies
  - Ohio Board of Regents

Outline

- Background in knowledge
  - Knowledge transfer
    - Learn by receiving, forget by sending
- Definition of Belief
  - Limits ensure it is well-defined
  - Belief transfer
    - Gain/lose belief by receiving or sending
- Examples:
  - Directories for distributed components
  - Asynchronous leases
- Conclusions

Remote Belief: Preserving Volition for Loosely Coupled Processes

Nuh Aydin (Kenyon College)

Paul Sivilotti (Ohio State University)

http://www.cis.ohio-state.edu/~paolo

Model of Computation

- Process computation
  - Finite sequence of events
  - Every process has a set of possible sequences
    - Prefix closed
- System computation
  - Finite sequence of events
  - Every projection is a process computation
  - Every receive has a corresponding send

$Q : (S_t | S_h)\ e_Q^*$
$P : e_P^*\ (R_t | R_h)\ e_P^*$

$x = \langle e_P\ S_h\ e_P\ e_P\ R_h\ e_P\ e_Q \rangle$