On the Use of Epistemic Ordering Functions as Decision Criteria for Automated and Assisted Belief Revision in SNePS
(Preliminary Report)

Ari I. Fogel and Stuart C. Shapiro

Department of Computer Science and Engineering
University at Buffalo, Buffalo, NY 14260
{arifogel,shapiro}@buffalo.edu

Outline

Goal

- Algorithms for using a user-supplied epistemic ordering relation
- for automated or user-assisted belief revision
- with a miminal burden on the user.
- Generalizes previous work
  on use of epistemic ordering for BR in SNePS.

Setting, Representation

- SNePS Knowledge Representation and Reasoning System.
- Implemented.
- First-Order Logic.
- Finite Belief Base (Knowledge Base, KB).
- Every belief either hypothesis (hyp) or derived (der). (Could be both.)

Setting, Representation

- SNePS Knowledge Representation and Reasoning System.
- Implemented.
- First-Order Logic.
- Finite Belief Base (Knowledge Base, KB).
- Every belief either hypothesis (hyp) or derived (der). (Could be both.)

Setting, Representation

- SNePS Knowledge Representation and Reasoning System.
- Implemented.
- First-Order Logic.
- Finite Belief Base (Knowledge Base, KB).
- Every belief either hypothesis (hyp) or derived (der).
(Could be both.)

Setting, Representation

- SNePS Knowledge Representation and Reasoning System.
- Implemented.
- First-Order Logic.
- Finite Belief Base (Knowledge Base, KB).
- Every belief either hypothesis (hyp) or derived (der). (Could be both.)

Setting, Representation

- SNePS Knowledge Representation and Reasoning System.
- Implemented.
- First-Order Logic.
- Finite Belief Base (Knowledge Base, KB).
- Every belief either hypothesis (hyp) or derived (der).
  (Could be both.)

Setting, Inference

- Forward, backward, and bi-directional inference.
- Uses Relevance Logic (R, paraconsistent).
- Every belief has a set of origin sets (OSs).
  - One OS for each way it has been derived **so far**.
- OS = set of hyps actually used for the derivation.
  - Computed by rules of inference.
  - If p is a hyp, $\{p\} \in os(p)$.
- Context = a set of hyps.
- Current Context (CC) = a set of hyps currently believed.
- Proposition p is asserted (believed) iff $\exists s[s \in os(p) \land s \subseteq CC]$.

[Martins & Shapiro, AIJ, 1988]

Setting, Inference

- Forward, backward, and bi-directional inference.
- Uses Relevance Logic (R, paraconsistent).
- Every belief has a set of origin sets (OSs).
  - One OS for each way it has been derived **so far**.
- OS = set of hyps actually used for the derivation.
  - Computed by rules of inference.
  - If p is a hyp, $\{p\} \in os(p)$.
- Context = a set of hyps.
- Current Context (CC) = a set of hyps currently believed.
- Proposition p is asserted (believed) iff $\exists s[s \in os(p) \wedge s \subseteq CC]$.

[Martins & Shapiro, AIJ, 1988]

Setting, Inference

- Forward, backward, and bi-directional inference.
- Uses Relevance Logic (R, paraconsistent).
- Every belief has a set of origin sets (OSs).
  - One OS for each way it has been derived **so far**.
- OS = set of hyps actually used for the derivation.
  - Computed by rules of inference.
  - If p is a hyp, $\{p\} \in os(p)$.
- Context = a set of hyps.
- Current Context (CC) = a set of hyps currently believed.
- Proposition p is asserted (believed) iff $\exists s[s \in os(p) \land s \subseteq CC]$.

[Martins & Shapiro, AIJ, 1988]

Setting, Inference

- Forward, backward, and bi-directional inference.
- Uses Relevance Logic (R, paraconsistent).
- Every belief has a set of origin sets (OSs).
  - One OS for each way it has been derived **so far**.
- OS = set of hyps actually used for the derivation.
  - Computed by rules of inference.
  - If p is a hyp, $\{p\} \in os(p)$.
- Context = a set of hyps.
- Current Context (CC) = a set of hyps currently believed.
- Proposition p is asserted (believed) iff $\exists s[s \in os(p) \wedge s \subseteq CC]$.

[Martins & Shapiro, AIJ, 1988]

Setting, Inference

- Forward, backward, and bi-directional inference.
- Uses Relevance Logic (R, paraconsistent).
- Every belief has a set of origin sets (OSs).
  - One OS for each way it has been derived **so far**.
- OS $=$ set of hyps actually used for the derivation.
  - Computed by rules of inference.
  - If p is a hyp, $\{p\} \in os(p)$.
- Context $=$ a set of hyps.
- Current Context (CC) $=$ a set of hyps currently believed.
- Proposition p is asserted (believed) iff $\exists s[s \in os(p) \wedge s \subseteq CC]$.

[Martins & Shapiro, AIJ, 1988]

Setting, Inference

- Forward, backward, and bi-directional inference.
- Uses Relevance Logic (R, paraconsistent).
- Every belief has a set of origin sets (OSs).
  - One OS for each way it has been derived **so far**.
- OS = set of hyps actually used for the derivation.
  - Computed by rules of inference.
  - If p is a hyp, $\{p\} \in os(p)$.
- Context = a set of hyps.
- Current Context (CC) = a set of hyps currently believed.
- Proposition p is asserted (believed) iff $\exists s[s \in os(p) \wedge s \subseteq CC]$.

[Martins & Shapiro, AIJ, 1988]

Setting, Inference

- Forward, backward, and bi-directional inference.
- Uses Relevance Logic (R, paraconsistent).
- Every belief has a set of origin sets (OSs).
  - One OS for each way it has been derived **so far**.
- OS = set of hyps actually used for the derivation.
  - Computed by rules of inference.
  - If p is a hyp, $\{p\} \in os(p)$.
- Context = a set of hyps.
- Current Context (CC) = a set of hyps currently believed.
- Proposition p is asserted (believed)
  iff $\exists s[s \in os(p) \land s \subseteq CC]$.

[Martins & Shapiro, AIJ, 1988]

SNeBR

- Contradiction recognized
  when both some p and $\neg p$ become asserted (believed).
  - Same data object used for p in both wffs.
  - Second one (call it $\neg p$) could have been
    - a hyp just added to the KB;
    - derived by forward inference from a hyp just added to the KB;
    - derived by backward inference
      from some hyps not previously realized to be inconsistent with p.
- Each of p, $\neg p$ could be a hypothesis or derived.
- Nogood $= s_1 \cup s_2 \; s.t. \; s_1 \in os(p) \wedge s_2 \in os(\neg p)$
  a minimally inconsistent set of hyps.
- To restore KB to state of not being known to be inconsistent,
  must remove one hyp in each nogood from CC.
  Guaranteed to be sufficient.

SNeBR

- Contradiction recognized
  when both some p and $\neg p$ become asserted (believed).
  - Same data object used for p in both wffs.
  - Second one (call it $\neg p$) could have been
    - a hyp just added to the KB;
    - derived by forward inference from a hyp just added to the KB;
    - derived by backward inference
      from some hyps not previously realized to be inconsistent with p.
- Each of p, $\neg p$ could be a hypothesis or derived.
- Nogood $= s_1 \cup s_2$ $s.t.$ $s_1 \in os(p) \wedge s_2 \in os(\neg p)$
  a minimally inconsistent set of hyps.
- To restore KB to state of not being known to be inconsistent,
  must remove one hyp in each nogood from CC.
  Guaranteed to be sufficient.

SNeBR

- Contradiction recognized
  when both some p and $\neg p$ become asserted (believed).
  - Same data object used for p in both wffs.
  - Second one (call it $\neg p$) could have been
    - a hyp just added to the KB;
    - derived by forward inference from a hyp just added to the KB;
    - derived by backward inference
      from some hyps not previously realized to be inconsistent with p.
- Each of p, $\neg p$ could be a hypothesis or derived.
- Nogood $= s_1 \cup s_2$ $s.t.$ $s_1 \in os(p) \wedge s_2 \in os(\neg p)$
  a minimally inconsistent set of hyps.
- To restore KB to state of not being known to be inconsistent,
  must remove one hyp in each nogood from CC.
  Guaranteed to be sufficient.

SNeBR

- Contradiction recognized
  when both some p and $\neg p$ become asserted (believed).
  - Same data object used for p in both wffs.
  - Second one (call it $\neg p$) could have been
    - a hyp just added to the KB;
    - derived by forward inference from a hyp just added to the KB;
    - derived by backward inference
      from some hyps not previously realized to be inconsistent with p.
- Each of p, $\neg p$ could be a hypothesis or derived.
- Nogood $= s_1 \cup s_2$ $s.t.$ $s_1 \in os(p) \wedge s_2 \in os(\neg p)$
  a minimally inconsistent set of hyps.
- To restore KB to state of not being known to be inconsistent,
  must remove one hyp in each nogood from CC.
  Guaranteed to be sufficient.

SNeBR

- Contradiction recognized
  when both some p and $\neg p$ become asserted (believed).
  - Same data object used for p in both wffs.
  - Second one (call it $\neg p$) could have been
    - a hyp just added to the KB;
    - derived by forward inference from a hyp just added to the KB;
    - derived by backward inference
      from some hyps not previously realized to be inconsistent with p.
- Each of p, $\neg p$ could be a hypothesis or derived.
- Nogood $= s_1 \cup s_2$ $s.t.$ $s_1 \in os(p) \wedge s_2 \in os(\neg p)$
  a minimally inconsistent set of hyps.
- To restore KB to state of not being known to be inconsistent,
  must remove one hyp in each nogood from CC.
  Guaranteed to be sufficient.

SNeBR

- Contradiction recognized
  when both some p and $\neg p$ become asserted (believed).
  - Same data object used for p in both wffs.
  - Second one (call it $\neg p$) could have been
    - a hyp just added to the KB;
    - derived by forward inference from a hyp just added to the KB;
    - derived by backward inference
      from some hyps not previously realized to be inconsistent with p.
- Each of p, $\neg p$ could be a hypothesis or derived.
- Nogood $= s_1 \cup s_2$ $s.t.$ $s_1 \in os(p) \wedge s_2 \in os(\neg p)$
  a minimally inconsistent set of hyps.
- To restore KB to state of not being known to be inconsistent,
  must remove one hyp in each nogood from CC.
  Guaranteed to be sufficient.

SNeBR

- Contradiction recognized
  when both some p and $\neg p$ become asserted (believed).
  - Same data object used for p in both wffs.
  - Second one (call it $\neg p$) could have been
    - a hyp just added to the KB;
    - derived by forward inference from a hyp just added to the KB;
    - derived by backward inference
      from some hyps not previously realized to be inconsistent with p.
- Each of p, $\neg p$ could be a hypothesis or derived.
- Nogood $= s_1 \cup s_2 \; s.t. \; s_1 \in os(p) \wedge s_2 \in os(\neg p)$
  a minimally inconsistent set of hyps.
- To restore KB to state of not being known to be inconsistent,
  must remove one hyp in each nogood from CC.
  Guaranteed to be sufficient.

SNeBR

- Contradiction recognized
  when both some p and $\neg p$ become asserted (believed).
  - Same data object used for p in both wffs.
  - Second one (call it $\neg p$) could have been
    - a hyp just added to the KB;
    - derived by forward inference from a hyp just added to the KB;
    - derived by backward inference
      from some hyps not previously realized to be inconsistent with p.
- Each of p, $\neg p$ could be a hypothesis or derived.
- Nogood $= s_1 \cup s_2 \ s.t. \ s_1 \in os(p) \wedge s_2 \in os(\neg p)$
  a minimally inconsistent set of hyps.
- To restore KB to state of not being known to be inconsistent,
  must remove one hyp in each nogood from CC.
  Guaranteed to be sufficient.

Assisted Belief Revision in SNeBR

- Present each nogood to user.
- Ask user to choose at least one hyp per nogood for removal from CC.
- Is non-prioritized belief revision.
  (Not predetermined whether p or $\neg p$ survives.)

Assisted Belief Revision in SNeBR

- Present each nogood to user.
- Ask user to choose at least one hyp per nogood for removal from CC.
- Is non-prioritized belief revision.
  (Not predetermined whether p or $\neg p$ survives.)

Assisted Belief Revision in SNeBR

- Present each nogood to user.
- Ask user to choose at least one hyp per nogood for removal from CC.
- Is non-prioritized belief revision.
  (Not predetermined whether p or $\neg p$ survives.)

Previous Restricted Prioritized BR in SNeBR

- In context of SNePS-based agents acting on-line.
- Assumes all beliefs are about the current state of the world.
- Agent performs `believe(p)` but currently believes `~p`.
  - If `nor{p, ...}` is believed as hyp, it is removed from CC.
  - If `xor{p, q, ...}` is believed, and q is believed as a hyp, q is removed from CC.
  - If `andor(0,1){p, q, ...}` is believed, and q is believed as a hyp, q is removed from CC.
  - Else do assisted BR.
- "State Constraints"

[Shapiro & Kandefer, NRAC-2005]

Previous Restricted Prioritized BR in SNeBR

- In context of SNePS-based agents acting on-line.
- Assumes all beliefs are about the current state of the world.
- Agent performs `believe(p)`
  but currently believes `~p`.
  - If `nor{p, ...}` is believed as hyp, it is removed from CC.
  - If `xor{p, q, ...}` is believed,
    and q is believed as a hyp,
    q is removed from CC.
  - If `andor(0,1){p, q, ...}` is believed,
    and q is believed as a hyp,
    q is removed from CC.
  - Else do assisted BR.
- "State Constraints"

[Shapiro & Kandefer, NRAC-2005]

Previous Restricted Prioritized BR in SNeBR

- In context of SNePS-based agents acting on-line.
- Assumes all beliefs are about the current state of the world.
- Agent performs `believe(p)`
  but currently believes `~p`.
  - If `nor{p, ...}` is believed as hyp, it is removed from CC.
  - If `xor{p, q, ...}` is believed,
    and q is believed as a hyp,
    q is removed from CC.
  - If `andor(0,1){p, q, ...}` is believed,
    and q is believed as a hyp,
    q is removed from CC.
  - Else do assisted BR.
- "State Constraints"

[Shapiro & Kandefer, NRAC-2005]

Previous Restricted Prioritized BR in SNeBR

- In context of SNePS-based agents acting on-line.
- Assumes all beliefs are about the current state of the world.
- Agent performs `believe(p)`
  but currently believes `~p`.
  - If `nor{p, ...}` is believed as hyp, it is removed from CC.
  - If `xor{p, q, ...}` is believed,
    and q is believed as a hyp,
    q is removed from CC.
  - If `andor(0,1){p, q, ...}` is believed,
    and q is believed as a hyp,
    q is removed from CC.
  - Else do assisted BR.
- "State Constraints"

[Shapiro & Kandefer, NRAC-2005]

Previous Restricted Prioritized BR in SNeBR

- In context of SNePS-based agents acting on-line.
- Assumes all beliefs are about the current state of the world.
- Agent performs `believe(p)`
  but currently believes `~p`.
  - If `nor{p, ...}` is believed as hyp, it is removed from CC.
  - If `xor{p, q, ...}` is believed,
    and q is believed as a hyp,
    q is removed from CC.
  - If `andor(0,1){p, q, ...}` is believed,
    and q is believed as a hyp,
    q is removed from CC.
  - Else do assisted BR.
- "State Constraints"

[Shapiro & Kandefer, NRAC-2005]

Previous Restricted Prioritized BR in SNeBR

- In context of SNePS-based agents acting on-line.
- Assumes all beliefs are about the current state of the world.
- Agent performs `believe(p)`
  but currently believes `~p`.
  - If `nor{p, ...}` is believed as hyp, it is removed from CC.
  - If `xor{p, q, ...}` is believed,
    and q is believed as a hyp,
    q is removed from CC.
  - If `andor(0,1){p, q, ...}` is believed,
    and q is believed as a hyp,
    q is removed from CC.
  - Else do assisted BR.
- "State Constraints"

[Shapiro & Kandefer, NRAC-2005]

Previous Restricted Prioritized BR in SNeBR

- In context of SNePS-based agents acting on-line.
- Assumes all beliefs are about the current state of the world.
- Agent performs `believe(p)`
  but currently believes `~p`.
  - If `nor{p, ...}` is believed as hyp, it is removed from CC.
  - If `xor{p, q, ...}` is believed,
    and q is believed as a hyp,
    q is removed from CC.
  - If `andor(0,1){p, q, ...}` is believed,
    and q is believed as a hyp,
    q is removed from CC.
  - Else do assisted BR.
- "State Constraints"

[Shapiro & Kandefer, NRAC-2005]

Previous Restricted Prioritized BR in SNeBR

- In context of SNePS-based agents acting on-line.
- Assumes all beliefs are about the current state of the world.
- Agent performs `believe(p)`
  but currently believes `~p`.
  - If `nor{p, ...}` is believed as hyp, it is removed from CC.
  - If `xor{p, q, ...}` is believed,
    and q is believed as a hyp,
    q is removed from CC.
  - If `andor(0,1){p, q, ...}` is believed,
    and q is believed as a hyp,
    q is removed from CC.
  - Else do assisted BR.
- "State Constraints"

[Shapiro & Kandefer, NRAC-2005]

Problem Statement

- Have
  - A set of nogoods, $\Sigma = \{\sigma_1, ..., \sigma_n\}$.
  - A set of prioritized beliefs, P, possibly empty.
  - total preorder, $\leq$, over hyps.
    - $\forall h_1, h_2 \in hyps, h_1 \leq h_2 \vee h_2 \leq h_1$.
    - transitive.
    - $\forall h_1, h_2[h_1 \in P \wedge h_2 \notin P \rightarrow h_1 > h_2]$
- Assume only moderate burden on user to specify $\leq$.
- Want
  - A set T of hyps to retract.
  - Retract at least one hyp from each nogood.
    $\forall \sigma[\sigma \in \Sigma \rightarrow \exists \tau[\tau \in (T \cap \sigma)]$
  - Don't retract w if could have chosen τ and $w > \tau$.
    $\forall \tau[\tau \in T \rightarrow \exists \sigma[\sigma \in \Sigma \wedge \tau \in \sigma \wedge \forall w[w \in \sigma \rightarrow \tau \leq w]]]$
  - Retract as few hyps as necessary.
    $\forall T'[T' \subset T \rightarrow \neg\forall \sigma[\sigma \in \Sigma \rightarrow \exists \tau[\tau \in (T' \cap \sigma)]]]$

Problem Statement

- Have
  - A set of nogoods, $\Sigma = \{\sigma_1, ..., \sigma_n\}$.
  - A set of prioritized beliefs, P, possibly empty.
  - total preorder, $\leq$, over hyps.
    - $\forall h_1, h_2 \in hyps, h_1 \leq h_2 \vee h_2 \leq h_1$.
    - transitive.
    - $\forall h_1, h_2 [h_1 \in P \wedge h_2 \notin P \rightarrow h_1 > h_2]$
- Assume only moderate burden on user to specify $\leq$.
- Want
  - A set T of hyps to retract.
  - Retract at least one hyp from each nogood.
    $\forall \sigma [\sigma \in \Sigma \rightarrow \exists \tau [\tau \in (T \cap \sigma)]]$
  - Don't retract w if could have chosen τ and $w > \tau$.
    $\forall \tau [\tau \in T \rightarrow \exists \sigma [\sigma \in \Sigma \wedge \tau \in \sigma \wedge \forall w [w \in \sigma \rightarrow \tau \leq w]]]$
  - Retract as few hyps as necessary.
    $\forall T' [T' \subset T \rightarrow \neg \forall \sigma [\sigma \in \Sigma \rightarrow \exists \tau [\tau \in (T' \cap \sigma)]]]$

Problem Statement

- Have
  - A set of nogoods, $\Sigma = \{\sigma_1, ..., \sigma_n\}$.
  - A set of prioritized beliefs, P, possibly empty.
  - total preorder, $\leq$, over hyps.
    - $\forall h_1, h_2 \in hyps, h_1 \leq h_2 \vee h_2 \leq h_1$.
    - transitive.
    - $\forall h_1, h_2[h_1 \in P \wedge h_2 \notin P \rightarrow h_1 > h_2]$
- Assume only moderate burden on user to specify $\leq$.
- Want
  - A set T of hyps to retract.
  - Retract at least one hyp from each nogood.
    $\forall \sigma[\sigma \in \Sigma \rightarrow \exists \tau[\tau \in (T \cap \sigma)]$
  - Don't retract w if could have chosen τ and $w > \tau$.
    $\forall \tau[\tau \in T \rightarrow \exists \sigma[\sigma \in \Sigma \wedge \tau \in \sigma \wedge \forall w[w \in \sigma \rightarrow \tau \leq w]]]$
  - Retract as few hyps as necessary.
    $\forall T'[T' \subset T \rightarrow \neg \forall \sigma[\sigma \in \Sigma \rightarrow \exists \tau[\tau \in (T' \cap \sigma)]]]$

Problem Statement

- Have
  - A set of nogoods, $\Sigma = \{\sigma_1, ..., \sigma_n\}$.
  - A set of prioritized beliefs, P, possibly empty.
  - total preorder, $\leq$, over hyps.
    - $\forall h_1, h_2 \in hyps, h_1 \leq h_2 \vee h_2 \leq h_1$.
    - transitive.
    - $\forall h_1, h_2[h_1 \in P \wedge h_2 \notin P \rightarrow h_1 > h_2]$
- Assume only moderate burden on user to specify $\leq$.
- Want
  - A set T of hyps to retract.
  - Retract at least one hyp from each nogood. $\forall \sigma[\sigma \in \Sigma \rightarrow \exists \tau[\tau \in (T \cap \sigma)]$
  - Don't retract w if could have chosen τ and $w > \tau$. $\forall \tau[\tau \in T \rightarrow \exists \sigma[\sigma \in \Sigma \wedge \tau \in \sigma \wedge \forall w[w \in \sigma \rightarrow \tau \leq w]]]$
  - Retract as few hyps as necessary. $\forall T'[T' \subset T \rightarrow \neg \forall \sigma[\sigma \in \Sigma \rightarrow \exists \tau[\tau \in (T' \cap \sigma)]]]$

Problem Statement

- Have
  - A set of nogoods, $\Sigma = \{\sigma_1, ..., \sigma_n\}$.
  - A set of prioritized beliefs, P, possibly empty.
  - total preorder, $\leq$, over hyps.
    - $\forall h_1, h_2 \in hyps, h_1 \leq h_2 \vee h_2 \leq h_1$.
    - transitive.
    - $\forall h_1, h_2[h_1 \in P \wedge h_2 \notin P \rightarrow h_1 > h_2]$
- Assume only moderate burden on user to specify $\leq$.
- Want
  - A set T of hyps to retract.
  - Retract at least one hyp from each nogood.
    $\forall \sigma[\sigma \in \Sigma \rightarrow \exists \tau[\tau \in (T \cap \sigma)]$
  - Don't retract w if could have chosen τ and $w > \tau$.
    $\forall \tau[\tau \in T \rightarrow \exists \sigma[\sigma \in \Sigma \wedge \tau \in \sigma \wedge \forall w[w \in \sigma \rightarrow \tau \leq w]]]$
  - Retract as few hyps as necessary.
    $\forall T'[T' \subset T \rightarrow \neg \forall \sigma[\sigma \in \Sigma \rightarrow \exists \tau[\tau \in (T' \cap \sigma)]]]$

Problem Statement

- Have
  - A set of nogoods, $\Sigma = \{\sigma_1, ..., \sigma_n\}$.
  - A set of prioritized beliefs, P, possibly empty.
  - total preorder, $\leq$, over hyps.
    - $\forall h_1, h_2 \in hyps, h_1 \leq h_2 \vee h_2 \leq h_1$.
    - transitive.
    - $\forall h_1, h_2[h_1 \in P \wedge h_2 \notin P \rightarrow h_1 > h_2]$
- Assume only moderate burden on user to specify $\leq$.
- Want
  - A set T of hyps to retract.
  - Retract at least one hyp from each nogood. $\forall\sigma[\sigma \in \Sigma \rightarrow \exists\tau[\tau \in (T \cap \sigma)]$
  - Don't retract w if could have chosen τ and $w > \tau$. $\forall\tau[\tau \in T \rightarrow \exists\sigma[\sigma \in \Sigma \wedge \tau \in \sigma \wedge \forall w[w \in \sigma \rightarrow \tau \leq w]]]$
  - Retract as few hyps as necessary. $\forall T'[T' \subset T \rightarrow \neg\forall\sigma[\sigma \in \Sigma \rightarrow \exists\tau[\tau \in (T' \cap \sigma)]]]$

Problem Statement

- Have
  - A set of nogoods, $\Sigma = \{\sigma_1, ..., \sigma_n\}$.
  - A set of prioritized beliefs, P, possibly empty.
  - total preorder, $\leq$, over hyps.
    - $\forall h_1, h_2 \in hyps, h_1 \leq h_2 \vee h_2 \leq h_1$.
    - transitive.
    - $\forall h_1, h_2[h_1 \in P \wedge h_2 \notin P \rightarrow h_1 > h_2]$
- Assume only moderate burden on user to specify $\leq$.
- Want
  - A set T of hyps to retract.
  - Retract at least one hyp from each nogood.
    $\forall \sigma[\sigma \in \Sigma \rightarrow \exists \tau[\tau \in (T \cap \sigma)]$
  - Don't retract w if could have chosen τ and $w > \tau$.
    $\forall \tau[\tau \in T \rightarrow \exists \sigma[\sigma \in \Sigma \wedge \tau \in \sigma \wedge \forall w[w \in \sigma \rightarrow \tau \leq w]]]$
  - Retract as few hyps as necessary.
    $\forall T'[T' \subset T \rightarrow \neg \forall \sigma[\sigma \in \Sigma \rightarrow \exists \tau[\tau \in (T' \cap \sigma)]]]$

Problem Statement

- Have
  - A set of nogoods, $\Sigma = \{\sigma_1, ..., \sigma_n\}$.
  - A set of prioritized beliefs, P, possibly empty.
  - total preorder, $\leq$, over hyps.
    - $\forall h_1, h_2 \in hyps, h_1 \leq h_2 \vee h_2 \leq h_1$.
    - transitive.
    - $\forall h_1, h_2[h_1 \in P \wedge h_2 \notin P \rightarrow h_1 > h_2]$
- Assume only moderate burden on user to specify $\leq$.
- Want
  - A set T of hyps to retract.
  - Retract at least one hyp from each nogood.
    $\forall \sigma[\sigma \in \Sigma \rightarrow \exists \tau[\tau \in (T \cap \sigma)]$
  - Don't retract w if could have chosen τ and $w > \tau$.
    $\forall \tau[\tau \in T \rightarrow \exists \sigma[\sigma \in \Sigma \wedge \tau \in \sigma \wedge \forall w[w \in \sigma \rightarrow \tau \leq w]]]$
  - Retract as few hyps as necessary.
    $\forall T'[T' \subset T \rightarrow \neg \forall \sigma[\sigma \in \Sigma \rightarrow \exists \tau[\tau \in (T' \cap \sigma)]]]$

Problem Statement

- Have
  - A set of nogoods, $\Sigma = \{\sigma_1, ..., \sigma_n\}$.
  - A set of prioritized beliefs, P, possibly empty.
  - total preorder, $\leq$, over hyps.
    - $\forall h_1, h_2 \in hyps, h_1 \leq h_2 \vee h_2 \leq h_1$.
    - transitive.
    - $\forall h_1, h_2[h_1 \in P \wedge h_2 \notin P \rightarrow h_1 > h_2]$
- Assume only moderate burden on user to specify $\leq$.
- Want
  - A set T of hyps to retract.
  - Retract at least one hyp from each nogood.
    $\forall \sigma[\sigma \in \Sigma \rightarrow \exists \tau[\tau \in (T \cap \sigma)]$
  - Don't retract w if could have chosen τ and $w > \tau$.
    $\forall \tau[\tau \in T \rightarrow \exists \sigma[\sigma \in \Sigma \wedge \tau \in \sigma \wedge \forall w[w \in \sigma \rightarrow \tau \leq w]]]$
  - Retract as few hyps as necessary.
    $\forall T'[T' \subset T \rightarrow \neg \forall \sigma[\sigma \in \Sigma \rightarrow \exists \tau[\tau \in (T' \cap \sigma)]]]$

Problem Statement

- Have
  - A set of nogoods, $\Sigma = \{\sigma_1, ..., \sigma_n\}$.
  - A set of prioritized beliefs, P, possibly empty.
  - total preorder, $\leq$, over hyps.
    - $\forall h_1, h_2 \in hyps, h_1 \leq h_2 \vee h_2 \leq h_1$.
    - transitive.
    - $\forall h_1, h_2[h_1 \in P \wedge h_2 \notin P \rightarrow h_1 > h_2]$
- Assume only moderate burden on user to specify $\leq$.
- Want
  - A set T of hyps to retract.
  - Retract at least one hyp from each nogood.
    $\forall \sigma[\sigma \in \Sigma \rightarrow \exists \tau[\tau \in (T \cap \sigma)]$
  - Don't retract w if could have chosen τ and $w > \tau$.
    $\forall \tau[\tau \in T \rightarrow \exists \sigma[\sigma \in \Sigma \wedge \tau \in \sigma \wedge \forall w[w \in \sigma \rightarrow \tau \leq w]]]$
  - Retract as few hyps as necessary.
    $\forall T'[T' \subset T \rightarrow \neg \forall \sigma[\sigma \in \Sigma \rightarrow \exists \tau[\tau \in (T' \cap \sigma)]]]$

Problem Statement

- Have
  - A set of nogoods, $\Sigma = \{\sigma_1, ..., \sigma_n\}$.
  - A set of prioritized beliefs, P, possibly empty.
  - total preorder, $\leq$, over hyps.
    - $\forall h_1, h_2 \in hyps, h_1 \leq h_2 \vee h_2 \leq h_1$.
    - transitive.
    - $\forall h_1, h_2[h_1 \in P \wedge h_2 \notin P \rightarrow h_1 > h_2]$
- Assume only moderate burden on user to specify $\leq$.
- Want
  - A set T of hyps to retract.
  - Retract at least one hyp from each nogood.
    $\forall \sigma[\sigma \in \Sigma \rightarrow \exists \tau[\tau \in (T \cap \sigma)]$
  - Don't retract w if could have chosen τ and $w > \tau$.
    $\forall \tau[\tau \in T \rightarrow \exists \sigma[\sigma \in \Sigma \wedge \tau \in \sigma \wedge \forall w[w \in \sigma \rightarrow \tau \leq w]]]$
  - Retract as few hyps as necessary.
    $\forall T'[T' \subset T \rightarrow \neg \forall \sigma[\sigma \in \Sigma \rightarrow \exists \tau[\tau \in (T' \cap \sigma)]]]$

In Case of Ties

If need to decide whether h_1 or h_2 goes into T
and $h_1 \leq h_2 \wedge h_2 \leq h_1$,
we have a tie that needs breaking.

3 Possibilities:

1. $\leq$ is a well preorder, and above doesn't occur.
2. Use $\lessdot_{\leq}$, a subset of $\leq$ that is a well preorder.
3. Ask the user, but as little as possible.

Algorithm 1 Using Well Preorder (Sketch)

Put minimally entrenched hyp first in every σ
Order Σ in descending order of first hyps of σs
while $(\Sigma \neq \emptyset)$ **do**
  Add first hyp of first σ to T
  Delete from Σ every σ that contains that hyp
end while

- Algorithm 1 is correct.
- Space complexity: $O(|\Sigma|)$ memory units.
- Time complexity: $O(|\Sigma|^2 \cdot |\sigma|_{max})$

See paper for proofs.

Algorithm 2 Using Total Preorder (Sketch)

loop
  for all $\sigma_i \in \Sigma$ **s.t.** σ_i has exactly one minimally entrenched hyp, p, AND the other hyps in σ_i are not minimally entrenched in any other σ **do**
    Add p to T, and delete from Σ every σ that contains p
    if $\Sigma = \emptyset$ **then return** T **end if**
  end for
  for some $\sigma \in \Sigma$ that has multiple minimally entrenched hyps **do**
    Query User which minimally entrenched hyp is least desired
    Modify $\leq$ accordingly
  end for
end loop

Algorithm 2 Analysis

- Algorithm 2 is correct.
- Space complexity: $O(|\Sigma|^2 \cdot |\sigma|^2_{max})$ memory units.
- Time complexity: $O(|\Sigma|^2 \cdot |\sigma|^2_{max})$

See paper for proofs.

Epistemic Ordering by Source Credibility

- Idea:
  Rank hypotheses by relative credibility of their sources.
- Based on:
  - Johnson & Shapiro, "Says Who?," UB TR 99-08
  - Shapiro & Johnson, "Automatic BR in SNePS," NMR-2000.
- Uses object-language meta-knowledge [Shapiro, *et al.*, AI Magazine, 2007]:
  - `HasSource(`p, s`)`: Belief p's source is s.
  - `IsBetterSource(`$s1, s2$`)`: Source $s1$ is more credible than source $s2$.
- $\leq$:
  - An unsourced belief is more entrenched than a sourced belief.
  - Two sourced beliefs are ordered based on the order of their sources.

Says Who KB

```
IsBetterSource(holybook, prof).

        IsBetterSource(prof, nerd).
        IsBetterSource(fran, nerd).

                IsBetterSource(nerd, sexist).

HasSource(all(x)(old(x)=>smart(x)), holybook).
HasSource(all(x)(grad(x)=>smart(x)), prof).
HasSource(all(x)(jock(x)=>~smart(x)), nerd).
HasSource(all(x)(female(x)=>~smart(x)), sexist).

HasSource(and{old(fran),grad(fran),jock(fran),female(fran)},fran).

: smart(fran)?
  wff24!:  smart(fran)
```

Lifting Restriction of Prioritized BR in SNeBR

- Revision of approach of SNePS Wumpus World Agent
  [Shapiro & Kandefer, NRAC-2005].
- Instead of state constraints being more entrenched,
  fluents are less entrenched.
- Uses meta-linguistic list of propositional fluent symbols.

Example of Using Fluents

```
: ^(setf *fluents* '(Facing))
(snepslog::Facing)

: xor{Facing(north),Facing(south),Facing(east),Facing(west)}.
  wff5!:  xor{Facing(west),Facing(east),Facing(south),Facing(north)}

: perform believe(Facing(west))

: Facing(?d)?
  wff9!:  ~Facing(north)
  wff8!:  ~Facing(south)
  wff7!:  ~Facing(east)
  wff4!:  Facing(west)

: perform believe(Facing(east))

: Facing(?d)?
  wff11!:  ~Facing(west)
  wff9!:  ~Facing(north)
  wff8!:  ~Facing(south)
  wff3!:  Facing(east)
```

Conclusions

In setting of

- Finite belief base
- Hypotheses identified
- Derived beliefs have (possibly multiple) origin sets
- Not all derivable beliefs have been derived
- Concern with known inconsistency (explicit contradiction)

Showed how to do

- Automatic prioritized or non-prioritized Belief Revision with a well preorder among hypotheses
- Minimally assisted prioritized or non-prioritized Belief Revision with a total preorder among hypotheses

Generalized several previous *ad hoc* techniques

For More Information

- Paper in the proceedings
- Ari's MS thesis:
  http://www.cse.buffalo.edu/sneps/Bibliography/fogelThesis.pdf