A Logic for Hybrid Rules

Axel Polleres

Universidad Rey Juan Carlos, Madrid

Joint work with: Jos de Bruijn, DERI Innsbruck,
David Pearce, Univ. Rey Juan Carlos,
Agustín Valverde, Univ. de Málaga

RuleML Workshop 2006
Outline

1. Motivation
2. Hybrid KBs
3. QEL
4. Embedding Hybrid KBs
5. Conclusions
Combine rules with negation as failure with classical theories:

- Defined for syntactically limited programs/FOL theories
- All give a modular definition of models by projection+reduct.

Questions:

- Can we generalize these combinations in a (non-classical) logic, i.e. with a non-modular model definition?
- Does this provide us with notions of equivalence commonly used (strong equivalence, uniform equivalence, etc.)?
Combine **rules with negation as failure** with **classical theories**:

- **Hybrid KB** approaches rely on (variants of) the Answer Set Semantics. [Rosati, 2005/2005b/2006, Heymans, et al. 2006]
- Defined for **syntactically limited** programs/FOL theories
- All give a **modular definition** of models by projection+reduct.

Questions:

- Can we generalize these combinations in a (non-classical) logic, i.e. with a non-modular model definition?
- Does this provide us with notions of equivalence commonly used (strong equivalence, uniform equivalence, etc.)?
Motivation – Hybrid Knowledge Bases

Combine **rules with negation as failure** with **classical theories**:

- **Hybrid KB** approaches rely on (variants of) the Answer Set Semantics. [Rosati, 2005/2005b/2006, Heymans, et al. 2006]
- Defined for **syntactically limited** programs/FOL theories
- All give a **modular definition** of models by projection+reduct.

Questions:

- Can we generalize these combinations in a (non-classical) logic, i.e. with a non-modular model definition?
- Does this provide us with notions of equivalence commonly used (strong equivalence, uniform equivalence, etc.)?
Combine **rules with negation as failure** with **classical theories**:

- **Hybrid KB** approaches rely on (variants of) the Answer Set Semantics. [Rosati, 2005/2005b/2006, Heymans, et al. 2006]
- Defined for **syntactically limited** programs/FOL theories
- All give a **modular definition** of models by projection+reduct.

**Questions:**

- Can we generalize these combinations in a (non-classical) logic, i.e. with a non-modular model definition?
- Does this provide us with notions of equivalence commonly used (strong equivalence, uniform equivalence, etc.)?
Combine **rules with negation as failure** with **classical theories**:

- **Hybrid KB** approaches rely on (variants of) the Answer Set Semantics. [Rosati, 2005/2005b/2006, Heymans, et al. 2006]
- Defined for **syntactically limited** programs/FOL theories
- All give a **modular definition** of models by projection + reduct.

Questions:

- Can we generalize these combinations in a (non-classical) logic, i.e. with a non-modular model definition?
- Does this provide us with notions of equivalence commonly used (strong equivalence, uniform equivalence, etc.)?
Combine rules with negation as failure with classical theories:

- **Hybrid KB** approaches rely on (variants of) the Answer Set Semantics. [Rosati, 2005/2005b/2006, Heymans, et al. 2006]
- Defined for **syntactically limited** programs/FOL theories
- All give a **modular definition** of models by projection + reduct.

Questions:

- Can we generalize these combinations in a (non-classical) logic, i.e. with a non-modular model definition?
- Does this provide us with notions of equivalence commonly used (strong equivalence, uniform equivalence, etc.)?
Hybrid Knowledge Bases – Generalized Definition

$\mathcal{K} = (\mathcal{T}, \mathcal{P})$ hybrid knowledge base:

- classical first-order theory $\mathcal{T}$ over function-free language
  $\mathcal{L}_{\mathcal{T}} = \langle C, P_{\mathcal{T}} \rangle$
- a logic program $\mathcal{P}$ over function-free language
  $\mathcal{L} = \langle C, P_{\mathcal{T}} \cup P_{\mathcal{P}} \rangle$, i.e. a set of rules:

  $a_1 \lor a_2 \lor \ldots \lor a_k \lor \neg a_{k+1} \lor \ldots \lor \neg a_l \leftarrow b_1, \ldots, b_m, \neg b_{m+1}, \ldots, \neg b_n$

  where $P_{\mathcal{T}} \cap P_{\mathcal{P}} = \emptyset$

Note:

- $\mathcal{T}$ and $\mathcal{P}$ talk about the same constants, and
- allowed predicate symbols in $\mathcal{P}$ are a superset of the predicate symbols in $\mathcal{L}_{\mathcal{T}}$. 
Hybrid Knowledge Bases – Generalized Definition

\( \mathcal{K} = (T, P) \) hybrid knowledge base:

- classical first-order theory \( T \) over function-free language \( L_T = \langle C, P_T \rangle \)
- a logic program \( P \) over function-free language \( L = \langle C, P_T \cup P_P \rangle \), i.e. a set of rules:

\[
\begin{align*}
a_1 \lor a_2 \lor \ldots \lor a_k \lor \neg a_{k+1} \lor \ldots \lor \neg a_l & \leftarrow b_1, \ldots, b_m, \neg b_{m+1}, \ldots, \neg b_n \\
\end{align*}
\]

where \( P_T \cap P_P = \emptyset \)

Note:

- \( T \) and \( P \) talk about the same constants, and
- allowed predicate symbols in \( P \) are a superset of the predicate symbols in \( L_T \).
Hybrid Knowledge Bases – Generalized Definition

\[ \mathcal{K} = (\mathcal{T}, \mathcal{P}) \] is a hybrid knowledge base:

- classical first-order theory \( \mathcal{T} \) over function-free language \( \mathcal{L}_\mathcal{T} = \langle C, P_\mathcal{T} \rangle \)
- a logic program \( \mathcal{P} \) over function-free language \( \mathcal{L} = \langle C, P_\mathcal{T} \cup P_\mathcal{P} \rangle \), i.e. a set of rules:

  \[
  a_1 \lor a_2 \lor \ldots \lor a_k \lor \neg a_{k+1} \lor \ldots \lor \neg a_l \leftarrow b_1, \ldots, b_m, \neg b_{m+1}, \ldots, \neg b_n
  \]

  where \( P_\mathcal{T} \cap P_\mathcal{P} = \emptyset \)

Note:

- \( \mathcal{T} \) and \( \mathcal{P} \) talk about the same constants, and
- allowed predicate symbols in \( \mathcal{P} \) are a superset of the predicate symbols in \( \mathcal{L}_\mathcal{T} \).
Hybrid Knowledge Bases – Generalized Definition

\[ \mathcal{K} = (\mathcal{T}, \mathcal{P}) \text{ hybrid knowledge base:} \]

- classical first-order theory \( \mathcal{T} \) over function-free language
  \[ \mathcal{L}_\mathcal{T} = \langle C, P_\mathcal{T} \rangle \]
- a logic program \( \mathcal{P} \) over function-free language
  \[ \mathcal{L} = \langle C, P_\mathcal{T} \cup P_\mathcal{P} \rangle \], i.e. a set of rules:
  \[
a_1 \lor a_2 \lor \ldots \lor a_k \lor \neg a_{k+1} \lor \ldots \lor \neg a_l \leftarrow b_1, \ldots, b_m, \neg b_{m+1}, \ldots, \neg b_n
  \]
  where \( P_\mathcal{T} \cap P_\mathcal{P} = \emptyset \)

Note:

- \( \mathcal{T} \) and \( \mathcal{P} \) talk about the same constants, and
- allowed predicate symbols in \( \mathcal{P} \) are a superset of the predicate symbols in \( \mathcal{L}_\mathcal{T} \).
Hybrid Knowledge Bases – Generalized Definition

\( \mathcal{K} = (\mathcal{T}, \mathcal{P}) \) hybrid knowledge base:

- classical first-order theory \( \mathcal{T} \) over function-free language 
  \( \mathcal{L}_\mathcal{T} = \langle C, P_\mathcal{T} \rangle \)
- a logic program \( \mathcal{P} \) over function-free language 
  \( \mathcal{L} = \langle C, P_\mathcal{T} \cup P_\mathcal{P} \rangle \), i.e. a set of rules:

  \[
  a_1 \lor a_2 \lor \ldots \lor a_k \lor \neg a_{k+1} \lor \ldots \lor \neg a_l \leftarrow b_1, \ldots, b_m, \neg b_{m+1}, \ldots, \neg b_n
  \]

  where \( P_\mathcal{T} \cap P_\mathcal{P} = \emptyset \)

Note:

- \( \mathcal{T} \) and \( \mathcal{P} \) talk about the same constants, and
- allowed predicate symbols in \( \mathcal{P} \) are a superset of the predicate symbols in \( \mathcal{L}_\mathcal{T} \).
Overall idea for a nonmonotonic semantics: “evaluate” \( P \) wrt a classical model of the theory and then compute stable models.

Let \( P \) be a ground program an \( I = \langle U, I \rangle \) an \( \mathcal{L} \)-structure, with

\[
U = (D, \sigma).
\]

\( \Pi(P, I) \), the projection of \( P \) wrt \( I \), obtained by

\begin{enumerate}
  \item deleting each rule with head literal \( p(t) \) (or \( \neg p(t) \)) over \( \text{At}_D(C, P_T) \) such that \( p(\sigma(t)) \in I \) (or \( p(\sigma(t)) \notin I \))
  \item deleting each rule with body literal \( p(t) \) (or \( \neg p(t) \)) over \( \text{At}_D(C, P_T) \) such that \( p(\sigma(t)) \notin I \) (or \( p(\sigma(t)) \in I \));
\end{enumerate}

and deleting occurrences of literals from \( \mathcal{L}_T \) from remaining rules.
Hybrid Knowledge Bases – Projection:

Overall idea for a nonmonotonic semantics: “evaluate” $\mathcal{P}$ wrt a classical model of the theory and then compute stable models.

Let $\mathcal{P}$ be a ground program and $\mathcal{I} = \langle U, I \rangle$ an $\mathcal{L}$-structure, with $U = (D, \sigma)$.

$\Pi(\mathcal{P}, \mathcal{I})$, the projection of $\mathcal{P}$ wrt $\mathcal{I}$, obtained by

1. deleting each rule with head literal $p(t)$ (or $\neg p(t)$) over $At_D(C, P_T)$ such that $p(\sigma(t)) \in I$ (or $p(\sigma(t)) \notin I$)

2. deleting each rule with body literal $p(t)$ (or $\neg p(t)$) over $At_D(C, P_T)$ such that $p(\sigma(t)) \notin I$ (or $p(\sigma(t)) \in I$);

and deleting occurrences of literals from $\mathcal{L}_T$ from remaining rules.
Overall idea for a nonmonotonic semantics: “evaluate” $\mathcal{P}$ wrt a classical model of the theory and then compute stable models.

Let $\mathcal{P}$ be a ground program and $\mathcal{I} = \langle U, I \rangle$ an $\mathcal{L}$-structure, with $U = (D, \sigma)$.

$\Pi(\mathcal{P}, \mathcal{I})$, the projection of $\mathcal{P}$ wrt $\mathcal{I}$, obtained by

1. deleting each rule with head literal $p(t)$ (or $\neg p(t)$) over $At_D(C, P_T)$ such that $p(\sigma(t)) \in I$ (or $p(\sigma(t)) \notin I$)

2. deleting each rule with body literal $p(t)$ (or $\neg p(t)$) over $At_D(C, P_T)$ such that $p(\sigma(t)) \notin I$ (or $p(\sigma(t)) \in I$);

and deleting occurrences of literals from $\mathcal{L}_T$ from remaining rules.
Hybrid Knowledge Bases – Projection:

Overall idea for a nonmonotonic semantics: “evaluate” \( P \) wrt a classical model of the theory and then compute stable models.

Let \( P \) be a ground program and \( I = \langle U, I \rangle \) an \( \mathcal{L} \)-structure, with \( U = (D, \sigma) \).

\( \Pi(P, I) \), the projection of \( P \) wrt \( I \), obtained by

1. deleting each rule with head literal \( p(t) \) (or \( \neg p(t) \)) over \( \text{At}_D(C, P_T) \) such that \( p(\sigma(t)) \in I \) (or \( p(\sigma(t)) \notin I \))
2. deleting each rule with body literal \( p(t) \) (or \( \neg p(t) \)) over \( \text{At}_D(C, P_T) \) such that \( p(\sigma(t)) \notin I \) (or \( p(\sigma(t)) \in I \));

and deleting occurrences of literals from \( \mathcal{L}_T \) from remaining rules.
Hybrid Knowledge Bases – Projection:

Overall idea for a nonmonotonic semantics: “evaluate” \( \mathcal{P} \) wrt a classical model of the theory and then compute stable models.

Let \( \mathcal{P} \) be a ground program an \( \mathcal{I} = \langle U, I \rangle \) an \( \mathcal{L} \)-structure, with \( U = (D, \sigma) \).

\( \Pi(\mathcal{P}, \mathcal{I}) \), the projection of \( \mathcal{P} \) wrt \( \mathcal{I} \), obtained by

1. deleting each rule with head literal \( p(t) \) (or \( \neg p(t) \)) over \( At_D(C, P_T) \) such that \( p(\sigma(t)) \in I \) (or \( p(\sigma(t)) \notin I \))

2. deleting each rule with body literal \( p(t) \) (or \( \neg p(t) \)) over \( At_D(C, P_T) \) such that \( p(\sigma(t)) \notin I \) (or \( p(\sigma(t)) \in I \));

and deleting occurrences of literals from \( \mathcal{L}_T \) from remaining rules.
Generalized Definition

NM-models

Example

Hybrid Knowledge Bases – NM-model:

Overall idea for a nonmonotonic semantics: “evaluate” \( \mathcal{P} \) wrt a classical model of the theory and then compute stable models.

Let \( \mathcal{K} = (\mathcal{T}, \mathcal{P}) \) be a hybrid knowledge base. An NM-model \( \mathcal{M} = \langle U, I \rangle \) of a hybrid knowledge base \( \mathcal{K} \) is a first-order \( \mathcal{L} \)-structure such that

1. \( \mathcal{M}|_{\mathcal{L}_\mathcal{T}} \) is a model of \( \mathcal{T} \) and
2. \( \mathcal{M}|_{\mathcal{L}_\mathcal{P}} \) is a stable model set of \( \Pi(gr_U(\mathcal{P}), \mathcal{M}|_{\mathcal{L}_\mathcal{T}}) \), i.e. \( \mathcal{M}|_{\mathcal{L}_\mathcal{P}} \) is a minimal Herbrand Model of the reduct \( \Pi(gr_U(\mathcal{P}), \mathcal{M})|_{\mathcal{L}_\mathcal{P}} \), obtained by taking all rules:
   - such that \( \mathcal{M}|_{\mathcal{L}_\mathcal{P}} \models a_i \) for negative head atoms \( a_i \) and
   - \( \mathcal{I} \not\models b_j \) for all negative body atoms \( b_j \).
Hybrid Knowledge Bases – NM-model:

Overall idea for a nonmonotonic semantics: “evaluate” \( \mathcal{P} \) wrt a classical model of the theory and then compute stable models.

Let \( \mathcal{K} = (\mathcal{T}, \mathcal{P}) \) be a hybrid knowledge base. An NM-model \( \mathcal{M} = \langle U, I \rangle \) of a hybrid knowledge base \( \mathcal{K} \) is a first-order \( \mathcal{L} \)-structure such that

1. \( \mathcal{M} \models_{\mathcal{L}_T} \) is a model of \( \mathcal{T} \) and
2. \( \mathcal{M} \models_{\mathcal{L}_P} \) is a stable model set of \( \Pi(gr_U(\mathcal{P}), \mathcal{M} \models_{\mathcal{L}_T}) \), i.e. \( \mathcal{M} \models_{\mathcal{L}_P} \) is a minimal Herbrand Model of the reduct \( \Pi(gr_U(\mathcal{P}), \mathcal{M}) \models_{\mathcal{L}_P} \), obtained by taking all rules:
   - such that \( \mathcal{M} \models_{\mathcal{L}_P} \models a_i \) for negative head atoms \( a_i \) and
   - \( \mathcal{I} \not\models b_j \) for all negative body atoms \( b_j \).
Overall idea for a nonmonotonic semantics: “evaluate” $\mathcal{P}$ wrt a classical model of the theory and then compute stable models.

Let $\mathcal{K} = (\mathcal{T}, \mathcal{P})$ be a hybrid knowledge base. An **NM-model** $\mathcal{M} = \langle U, I \rangle$ of a hybrid knowledge base $\mathcal{K}$ is a first-order $\mathcal{L}$-structure such that

1. $\mathcal{M}|_{\mathcal{L}_T}$ is a model of $\mathcal{T}$ and
2. $\mathcal{M}|_{\mathcal{L}_P}$ is a stable model set of $\Pi(\text{gr}_U(\mathcal{P}), \mathcal{M}|_{\mathcal{L}_T})$, i.e. $\mathcal{M}|_{\mathcal{L}_P}$ is a minimal Herbrand Model of the reduct $\Pi(\text{gr}_U(\mathcal{P}), \mathcal{M})^\mathcal{M}|_{\mathcal{L}_P}$, obtained by taking all rules:
   - such that $\mathcal{M}|_{\mathcal{L}_P} \models a_i$ for negative head atoms $a_i$ and
   - $\mathcal{I} \not\models b_j$ for all negative body atoms $b_j$. 

Axel Polleres
A Logic for Hybrid Rules
Motivation
Hybrid KBs
QEL
Embedding Hybrid KBs
Conclusions

Hybrid Knowledge Bases – NM-model:

Overall idea for a nonmonotonic semantics: “evaluate” $\mathcal{P}$ wrt a classical model of the theory and then compute stable models.

Let $\mathcal{K} = (\mathcal{T}, \mathcal{P})$ be a hybrid knowledge base. An **NM-model** $\mathcal{M} = \langle U, I \rangle$ of a hybrid knowledge base $\mathcal{K}$ is a first-order $\mathcal{L}$-structure such that

1. $\mathcal{M}|_{\mathcal{L}_T}$ is a model of $\mathcal{T}$ and
2. $\mathcal{M}|_{\mathcal{L}_P}$ is a stable model set of $\Pi(gr_U(\mathcal{P}), \mathcal{M}|_{\mathcal{L}_T})$, i.e. $\mathcal{M}|_{\mathcal{L}_P}$ is a minimal Herbrand Model of the reduction $\Pi(gr_U(\mathcal{P}), \mathcal{M})|_{\mathcal{L}_P}$, obtained by taking all rules:
   - such that $\mathcal{M}|_{\mathcal{L}_P} \models a_i$ for negative head atoms $a_i$ and
   - $\mathcal{I} \models b_j$ for all negative body atoms $b_j$. 

---

6 Axel Polleres

A Logic for Hybrid Rules
Overall idea for a nonmonotonic semantics: “evaluate” $\mathcal{P}$ wrt a classical model of the theory and then compute stable models.

Let $\mathcal{K} = (\mathcal{T}, \mathcal{P})$ be a hybrid knowledge base. An **NM-model** $\mathcal{M} = \langle U, I \rangle$ of a hybrid knowledge base $\mathcal{K}$ is a first-order $\mathcal{L}$-structure such that

1. $\mathcal{M}|_{\mathcal{L}_T}$ is a model of $\mathcal{T}$ and
2. $\mathcal{M}|_{\mathcal{L}_P}$ is a stable model set of $\Pi(g_{rU}(\mathcal{P}), \mathcal{M}|_{\mathcal{L}_T})$, i.e. $\mathcal{M}|_{\mathcal{L}_P}$ is a minimal Herbrand Model of the reduct $\Pi(g_{rU}(\mathcal{P}), \mathcal{M})^{\mathcal{M}|_{\mathcal{L}_P}}$, obtained by taking all rules:
   - such that $\mathcal{M}|_{\mathcal{L}_P} \models a_i$ for negative head atoms $a_i$ and
   - $I \not\models b_j$ for all negative body atoms $b_j$.  

6 Axel Polleres  A Logic for Hybrid Rules
Example – a small Hybrid KB:

Let $\mathcal{K} = (\mathcal{T}, \mathcal{P})$ with

**$\mathcal{T}$**: Each foaf:Person is a foaf:Agent:

$\forall x. PERSON(x) \rightarrow AGENT(x)$

AGENT(David)

**$\mathcal{P}$**: Some nonmonotonic rule on top

$PERSON(x) \leftarrow pcmember(x, LPNMR), AGENT(x), \neg \text{machine}(x)$

pcmember(David, LPNMR)

Is David a PERSON?

Recall: $\neg$ is “negation as failure” here!
Example – a small Hybrid KB:

Let $\mathcal{K} = (\mathcal{T}, \mathcal{P})$ with

$\mathcal{T}$: Each foaf:Person is a foaf:Agent:

$$\forall x. \text{PERSON}(x) \rightarrow \text{AGENT}(x)$$

$\text{AGENT}(\text{David})$

$\mathcal{P}$: Some nonmonotonic rule on top

$$\text{PERSON}(x) \leftarrow \text{pcmember}(x, \text{LPNMR}), \text{AGENT}(x), \neg \text{machine}(x)$$

$\text{pcmember}(\text{David}, \text{LPNMR})$

Is David a PERSON?

Recall: $\neg$ is “negation as failure” here!
Example - NM-models:

Classical models of $T$:

$\forall x. PERSON(x) \rightarrow AGENT(x)$

$AGENT(David)$

$M_1|_{L_T} = \{AGENT(David), \neg PERSON(David), \neg AGENT(LP NMR), \ldots \}$

$M_2|_{L_T} = \{AGENT(David), PERSON(David), \neg AGENT(LP NMR), \ldots \}$

$gr_U(\mathcal{P})$

$PERSON(David) \leftarrow$

$pcmember(David, LP NMR), AGENT(David), \neg machine(David)$

$PERSON(LP NMR) \leftarrow$

$pcmember(LP NMR, LP NMR), AGENT(LP NMR), \neg machine(LP NMR)$

$pcmember(David, LP NMR)$

Is David a PERSON?

$\Pi(gr_U(\mathcal{P}), M_2|_{L_T})$
Example - NM-models:

**Classical models of** $\mathcal{T}$:

$$\forall x. \text{PERSON}(x) \rightarrow \text{AGENT}(x)$$

$\text{AGENT}(\text{David})$

$$\mathcal{M}_1|_{\mathcal{L}_\mathcal{T}} = \{ \text{AGENT}(\text{David}), \neg \text{PERSON}(\text{David}), \neg \text{AGENT}(\text{LPNMR}), \ldots \}$$

$$\mathcal{M}_2|_{\mathcal{L}_\mathcal{T}} = \{ \text{AGENT}(\text{David}), \text{PERSON}(\text{David}), \neg \text{AGENT}(\text{LPNMR}), \ldots \}$$

$$\Pi(\text{gr}_U(\mathcal{P}), \mathcal{M}_1|_{\mathcal{L}_\mathcal{T}})$$

← $\text{pcmember}(\text{David}, \text{LPNMR}), \neg \text{machine}(\text{David})$.

$\text{pcmember}(\text{David}, \text{LPNMR})$

No stable models!

Is David a PERSON?

$$\Pi(\text{gr}_U(\mathcal{P}), \mathcal{M}_2|_{\mathcal{L}_\mathcal{T}})$$
Example - NM-models:

Classical models of $\mathcal{T}$:

$$\forall x.\text{PERSON}(x) \rightarrow \text{AGENT}(x)$$

$$\text{AGENT}(David)$$

$$\mathcal{M}_1|_{\mathcal{L}_\mathcal{T}} = \{ \text{AGENT}(David), \neg\text{PERSON}(David), \neg\text{AGENT}(LPNMR), \ldots \}$$

$$\mathcal{M}_2|_{\mathcal{L}_\mathcal{T}} = \{ \text{AGENT}(David), \text{PERSON}(David), \neg\text{AGENT}(LPNMR), \ldots \}$$

$$\Pi(\text{gr}_U(\mathcal{P}), \mathcal{M}_2|_{\mathcal{L}_\mathcal{T}})$$

$\text{pcmember}(David, LPNMR)$

One stable model…

Is David a PERSON? Yes!

$\text{PERSON}(David)$ in all NM-models, i.e. $\mathcal{K} \models_{\text{NM}} \text{PERSON}(David)$
Example - NM-models:

**Classical models of $\mathcal{T}$:**

\[
\forall x. \text{PERSON}(x) \rightarrow \text{AGENT}(x) \\
\text{AGENT}(\text{David})
\]

\[
\mathcal{M}_1|_{\mathcal{L}_\mathcal{T}} = \{\text{AGENT}(\text{David}), \neg \text{PERSON}(\text{David}), \neg \text{AGENT}(\text{LPNMR}), \ldots\}
\]

\[
\mathcal{M}_2|_{\mathcal{L}_\mathcal{T}} = \{\text{AGENT}(\text{David}), \text{PERSON}(\text{David}), \neg \text{AGENT}(\text{LPNMR}), \ldots\}
\]

\[
\Pi(gr_{U}(\mathcal{P}), \mathcal{M}_2|_{\mathcal{L}_\mathcal{T}})
\]

\[
\text{pmember}(\text{David, LPNMR})
\]

One stable model...

Is David a PERSON? Yes!

\[
\text{PERSON}(\text{David}) \text{ in all NM-models, i.e. } \mathcal{K} \models_{\text{NM}} \text{PERSON}(\text{David})
\]
Back to our question for a non-classical logic which covers this...  

- *Equilibrium logic* (Pearce, 1997) generalizes stable model semantics and answer set semantics for logic programs to arbitrary propositional theories.
- It is a nonmonotonic extension of the logic of *Here-and-there* (with strong negation).
- Model theory based on Kripke semantics for intuitionistic logic
- We need a first-order version here...
Back to our question for a non-classical logic which covers this... 

- *Equilibrium logic* (Pearce, 1997) generalizes stable model semantics and answer set semantics for logic programs to arbitrary propositional theories.
- It is a nonmonotonic extension of the logic of *Here-and-there* (with strong negation).
- Model theory based on Kripke semantics for intuitionistic logic
- We need a first-order version here...
Quantified Here-and-there Logic

- QHTs is complete for linear Kripke frames with two worlds “here” and “there” with a “static” domain over both worlds: \( h \leq t \).
- here-and-there structures: \( M = \langle (D, \sigma), I_h, I_t \rangle \)
- \( I_h, I_t \) are first-order-interpretations over \( D \) such that \( I_h \subseteq I_t \).

The models are extended to all formulas via the rules known in intuitionistic logic, notions of validity and logical consequence relation are the ones for (intuitionistic) Kripke semantics.
Quantified Here-and-there Logic

- **QHT$^s$** is complete for linear Kripke frames with two worlds “here” and “there” with a “static” domain over both worlds: $h \leq t$.
- **here-and-there structures**: $\mathcal{M} = \langle (D, \sigma), I_h, I_t \rangle$
- $I_h, I_t$ are first-order-interpretations over $D$ such that $I_h \subseteq I_t$.

The models are extended to all formulas via the rules known in intuitionistic logic, notions of validity and logical consequence relation are the ones for (intuitionistic) Kripke semantics.
For $w \in \{h, t\}$:

- $M, w \models \varphi \land \psi$ iff $M, w \models \varphi$ and $M, w \models \psi$.
- $M, w \models \varphi \lor \psi$ iff $M, w \models \varphi$ or $M, w \models \psi$.
- $M, t \models \varphi \rightarrow \psi$ iff $M, t \not\models \varphi$ or $M, t \models \psi$.
- $M, h \models \varphi \rightarrow \psi$ iff $M, t \not\models \varphi \rightarrow \psi$ and $M, h \not\models \varphi$ or $M, h \models \psi$.
- $M, w \models \neg \varphi$ iff $M, t \not\models \varphi$.
- $M, t \models \forall x \varphi(x)$ iff $M, t \models \varphi(d)$ for all $d \in D$.
- $M, h \models \forall x \varphi(x)$ iff $M, t \models \forall x \varphi(x)$ and $M, h \not\models \varphi(d)$ for all $d \in D$.
- $M, w \models \exists x \varphi(x)$ iff $M, w \models \varphi(d)$ for some $d \in D$. 
Quantified Equilibrium Logic (QEL)

- We write QHT*-structures more briefly as ordered pairs of atoms $\langle H, T \rangle$, with $H \subseteq T$.
- An QHT*-Structure $\langle H, T \rangle$ is said to be total if $H = T$.
- Order relation: $\langle H, T \rangle \preceq \langle H', T' \rangle$ if $T = T'$ and $H \subseteq H'$.
- $\langle H, T \rangle$ is an equilibrium model of $\Pi$ if is
  (i) $\langle H, T \rangle$ minimal under $\preceq$, and
  (ii) $\langle H, T \rangle$ is total.

QEL is determined by the equilibrium models of a theory.
• We write QHT$^s$-structures more briefly as ordered pairs of atoms $\langle H, T \rangle$, with $H \subseteq T$.

• An QHT$^s$-Structure $\langle H, T \rangle$ is said to be total if $H = T$

• Order relation: $\langle H, T \rangle \trianglelefteq \langle H', T' \rangle$ if $T = T'$ and $H \subseteq H'$

• $\langle H, T \rangle$ is an equilibrium model of $\Pi$ if is
  (i) $\langle H, T \rangle$ minimal under $\trianglelefteq$, and
  (ii) $\langle H, T \rangle$ is total.

QEL is determined by the equilibrium models of a theory.
Quantified Equilibrium Logic (QEL)

- We write QHT\(^s\)-structures more briefly as ordered pairs of atoms \(\langle H, T \rangle\), with \(H \subseteq T\).

- An QHT\(^s\)-Structure \(\langle H, T \rangle\) is said to be **total** if \(H = T\).

- Order relation: \(\langle H, T \rangle \leq \langle H', T' \rangle\) if \(T = T'\) and \(H \subseteq H'\).

- \(\langle H, T \rangle\) is an **equilibrium model** of \(\Pi\) if is
  1. \(\langle H, T \rangle\) minimal under \(\sqsubseteq\), and
  2. \(\langle H, T \rangle\) is total.

QEL is determined by the equilibrium models of a theory.
Quantified Equilibrium Logic (QEL)

- We write QHT\(^s\)-structures more briefly as ordered pairs of atoms \(\langle H, T \rangle\), with \(H \subseteq T\).

- An QHT\(^s\)-Structure \(\langle H, T \rangle\) is said to be total if \(H = T\).

- Order relation: \(\langle H, T \rangle \preceq \langle H', T' \rangle\) if \(T = T'\) and \(H \subseteq H'\).

- \(\langle H, T \rangle\) is an equilibrium model of \(\Pi\) if is
  1. \(\langle H, T \rangle\) minimal under \(\preceq\), and
  2. \(\langle H, T \rangle\) is total.

QEL is determined by the equilibrium models of a theory.
Quantified Equilibrium Logic (QEL)

- We write QHT$^s$-structures more briefly as ordered pairs of atoms $\langle H, T \rangle$, with $H \subseteq T$.
- An QHT$^s$-Structure $\langle H, T \rangle$ is said to be total if $H = T$.
- Order relation: $\langle H, T \rangle \trianglelefteq \langle H', T' \rangle$ if $T = T'$ and $H \subseteq H'$.
- $\langle H, T \rangle$ is an equilibrium model of $\Pi$ if is
  (i) $\langle H, T \rangle$ minimal under $\trianglelefteq$, and
  (ii) $\langle H, T \rangle$ is total.

QEL is determined by the equilibrium models of a theory.
Quantified Equilibrium Logic and Answer Set Semantics

- Equilibrium Logic generalizes Answer Set semantics for arbitrary formulae (including disjunctive and nested programs)

- Any rule

\[ a_1 \lor a_2 \lor \ldots \lor a_k \lor \neg a_{k+1} \lor \ldots \lor \neg a_l \leftarrow b_1, \ldots, b_m, \neg b_{m+1}, \ldots, \neg b_n \]

is just treated as (universally closed) formula in QEL:

\[ (\forall) a_1 \lor a_2 \lor \ldots \lor a_k \lor \neg a_{k+1} \lor \ldots \lor \neg a_l \leftarrow b_1 \land \ldots \land b_m \land \neg b_{m+1} \land \ldots \land \neg b_n \]

- Equilibrium models correspond to (open) answer sets: \( \langle T, T \rangle \) is an equilibrium model of \( \mathcal{P} \) iff \( T \) is an answer set of \( \Pi \).

- So, for \( \mathcal{K} = (\emptyset, P) \) answer sets and QEL-models correspond!

Quantified Equilibrium Logic and Answer Set Semantics

- Equilibrium Logic generalizes Answer Set semantics for arbitrary formulae (including disjunctive and nested programs)
- Any rule

\[ a_1 \lor a_2 \lor \ldots \lor a_k \lor \neg a_{k+1} \lor \ldots \lor \neg a_l \leftarrow b_1, \ldots, b_m, \neg b_{m+1}, \ldots, \neg b_n \]

is just treated as (universally closed) formula in QEL:

\[ (\forall) a_1 \lor a_2 \lor \ldots \lor a_k \lor \neg a_{k+1} \lor \ldots \lor \neg a_l \leftarrow b_1 \land \ldots \land b_m \land \neg b_{m+1} \land \ldots \land \neg b_n \]

- Equilibrium models correspond to (open) answer sets: \( \langle T, T \rangle \) is a equilibrium model of \( \mathcal{P} \) iff \( T \) is an answer set of \( \Pi \).
- So, for \( \mathcal{K} = (\emptyset, P) \) answer sets and QEL-models correspond!

**Quantified Equilibrium Logic and Answer Set Semantics**

- Equilibrium Logic generalizes Answer Set semantics for arbitrary formulae (including disjunctive and nested programs).
- Any rule

\[
a_1 \lor a_2 \lor \ldots \lor a_k \lor \neg a_{k+1} \lor \ldots \lor \neg a_l \leftarrow b_1, \ldots, b_m, \neg b_{m+1}, \ldots, \neg b_n
\]

is just treated as (universally closed) formula in QEL:

\[
(\forall) a_1 \lor a_2 \lor \ldots \lor a_k \lor \neg a_{k+1} \lor \ldots \lor \neg a_l \leftarrow b_1 \land \ldots \land b_m \land \neg b_{m+1} \land \ldots \land \neg b_n
\]

- Equilibrium models correspond to (open) answer sets: \(\langle T, T \rangle\) is a equilibrium model of \(\mathcal{P}\) iff \(T\) is an answer set of \(\Pi\).
- So, for \(\mathcal{K} = (\emptyset, P)\) answer sets and QEL-models correspond!

Quantified Equilibrium Logic and Answer Set Semantics

- Equilibrium Logic generalizes Answer Set semantics for arbitrary formulae (including disjunctive and nested programs)

- Any rule

\[ a_1 \lor a_2 \lor \ldots \lor a_k \lor \neg a_{k+1} \lor \ldots \lor \neg a_l \leftarrow b_1, \ldots, b_m, \neg b_{m+1}, \ldots, \neg b_n \]

is just treated as (universally closed) formula in QEL:

\[ (\forall) a_1 \lor a_2 \lor \ldots \lor a_k \lor \neg a_{k+1} \lor \ldots \lor \neg a_l \leftarrow b_1 \land \ldots \land b_m \land \neg b_{m+1} \land \ldots \land \neg b_n \]

- Equilibrium models correspond to (open) answer sets: \( \langle T, T \rangle \) is an equilibrium model of \( \mathcal{P} \) iff \( T \) is an answer set of \( \Pi \).

- So, for \( \mathcal{K} = (\emptyset, P) \) answer sets and QEL-models correspond!

Quantified Equilibrium Logic and Answer Set Semantics

- Equilibrium Logic generalizes Answer Set semantics for arbitrary formulae (including disjunctive and nested programs)

- Any rule

\[ a_1 \lor a_2 \lor \ldots \lor a_k \lor \neg a_{k+1} \lor \ldots \lor \neg a_l \leftarrow b_1, \ldots, b_m, \neg b_{m+1}, \ldots, \neg b_n \]

is just treated as (universally closed) formula in QEL:

\[ (\forall) a_1 \lor a_2 \lor \ldots \lor a_k \lor \neg a_{k+1} \lor \ldots \lor \neg a_l \leftarrow b_1 \land \ldots \land b_m \land \neg b_{m+1} \land \ldots \land \neg b_n \]

- Equilibrium models correspond to (open) answer sets: \( \langle T, T \rangle \) is a equilibrium model of \( \mathcal{P} \) iff \( T \) is an answer set of \( \Pi \).

- So, for \( \mathcal{K} = (\emptyset, P) \) answer sets and QEL-models correspond!

Quantified Equilibrium Logic and Answer Set Semantics

- Equilibrium Logic generalizes Answer Set semantics for arbitrary formulae (including disjunctive and nested programs)
- Any rule

\[ a_1 \lor a_2 \lor \ldots \lor a_k \lor \neg a_{k+1} \lor \ldots \lor \neg a_l \leftarrow b_1, \ldots, b_m, \neg b_{m+1}, \ldots, \neg b_n \]

is just treated as (universally closed) formula in QEL:

\[ (\forall) a_1 \lor a_2 \lor \ldots \lor a_k \lor \neg a_{k+1} \lor \ldots \lor \neg a_l \leftarrow b_1 \land \ldots \land b_m \land \neg b_{m+1} \land \ldots \land \neg b_n \]

- Equilibrium models correspond to (open) answer sets: \( \langle T, T \rangle \) is an equilibrium model of \( P \) iff \( T \) is an answer set of \( \Pi \).
- So, for \( K = (\emptyset, P) \) answer sets and QEL-models correspond!

Quantified Equilibrium Logic and Answer Set Semantics

- Equilibrium Logic generalizes Answer Set semantics for arbitrary formulae (including disjunctive and nested programs)

- Any rule

\[ a_1 \lor a_2 \lor \ldots \lor a_k \lor \neg a_{k+1} \lor \ldots \lor \neg a_l \leftarrow b_1, \ldots, b_m, \neg b_{m+1}, \ldots, \neg b_n \]

is just treated as (universally closed) formula in QEL:

\[
(\forall) a_1 \lor a_2 \lor \ldots \lor a_k \lor \neg a_{k+1} \lor \ldots \lor \neg a_l \leftarrow b_1 \land \ldots \land b_m \land \neg b_{m+1} \land \ldots \land \neg b_n
\]

- Equilibrium models correspond to (open) answer sets: \( \langle T, T \rangle \) is an equilibrium model of \( \mathcal{P} \) iff \( T \) is an answer set of \( \Pi \).

- So, for \( \mathcal{K} = (\emptyset, \mathcal{P}) \) answer sets and QEL-models correspond!

Embedding Hybrid Knowledge Bases

Q: Does the correspondence extend to hybrid KBs? Yes!

Idea: define embedding based on the observation that adding LEM makes intuitionistic logic classical!

Given a hybrid KB $\mathcal{K} = (\mathcal{T}, \mathcal{P})$ we call $\mathcal{T} \cup \text{st}(\mathcal{T}) \cup \mathcal{P}$ the stable closure of $\mathcal{K}$, where $\text{st}(\mathcal{T}) = \{ \forall x (p(x) \lor \neg p(x)) : p \in \mathcal{L}_T \}$.

Wake up! Main theorem of the paper!!! ;-(

Theorem

Let $\mathcal{K} = (\mathcal{T}, \mathcal{P})$ be a hybrid knowledge base. Let $\mathcal{M} = \langle U, T, T \rangle$ be a total here-and-there model of the stable closure of $\mathcal{K}$. Then $\mathcal{M}$ is an equilibrium model if and only if it is an NM-model of $\mathcal{K}$. 
Embedding Hybrid Knowledge Bases

Q: Does the correspondence extend to hybrid KBs? Yes!

Idea: define embedding based on the observation that adding LEM makes intuitionistic logic classical!

Given a hybrid KB $\mathcal{K} = (\mathcal{T}, \mathcal{P})$ we call $\mathcal{T} \cup \text{st}(\mathcal{T}) \cup \mathcal{P}$ the stable closure of $\mathcal{K}$, where $\text{st}(\mathcal{T}) = \{ \forall x (p(x) \lor \neg p(x)) : p \in \mathcal{L}_\mathcal{T} \}$.

Wake up! Main theorem of the paper!!! ;-)
Motivation
Hybrid KBs
QEL
Embedding Hybrid KBs
Conclusions

Embedding Hybrid Knowledge Bases

Q: Does the correspondence extend to hybrid KBs? Yes!

Idea: define embedding based on the observation that adding LEM makes intuitionistic logic classical!

Given a hybrid KB $\mathcal{K} = (\mathcal{T}, \mathcal{P})$ we call $\mathcal{T} \cup st(\mathcal{T}) \cup \mathcal{P}$ the stable closure of $\mathcal{K}$, where $st(\mathcal{T}) = \{\forall x(p(x) \vee \neg p(x)) : p \in \mathcal{L}_\mathcal{T}\}$.

Wake up! Main theorem of the paper!!! ;-)

Theorem

Let $\mathcal{K} = (\mathcal{T}, \mathcal{P})$ be a hybrid knowledge base. Let $\mathcal{M} = \langle U, T, T \rangle$ be a total here-and-there model of the stable closure of $\mathcal{K}$. Then $\mathcal{M}$ is an equilibrium model if and only if it is an NM-model of $\mathcal{K}$. 
Q: Does the correspondence extend to hybrid KBs? Yes!

**Idea:** define embedding based on the observation that adding LEM makes intuitionistic logic classical!

Given a hybrid KB $\mathcal{K} = (T, \mathcal{P})$ we call $T \cup \text{st}(T) \cup \mathcal{P}$ the stable closure of $\mathcal{K}$, where $\text{st}(T) = \{ \forall x (p(x) \lor \neg p(x)) : p \in \mathcal{L}_T \}$.

Wake up! Main theorem of the paper!!! ;-)

**Theorem**

Let $\mathcal{K} = (T, \mathcal{P})$ be a hybrid knowledge base. Let $\mathcal{M} = \langle U, T, T \rangle$ be a total here-and-there model of the stable closure of $\mathcal{K}$. Then $\mathcal{M}$ is an equilibrium model if and only if it is an NM-model of $\mathcal{K}$. 
Example - stable closure of $\mathcal{K}$:

\[
st(\mathcal{K}) = \mathcal{T} \cup st(\mathcal{T}) \cup \mathcal{P}
\]

\[
\forall x. PERSON(x) \rightarrow AGENT(x) \\
AGENT(David) \\
\forall x. PERSON(x) \lor \neg PERSON(x) \\
\forall x. AGENT(x) \lor \neg AGENT(x) \\
\forall x. PERSON(x) \leftarrow pcmember(x, LPNMR) \land AGENT(x) \land \neg machine(x) \\
pcmembver(David, LPNMR)
\]

There IS a classical model of this theory
\[
\mathcal{M} = \{ \neg PERSON(David), machine(David), \ldots \}
\]

Thus:
\[
K \not\models_{FOL} PERSON(David)
\]
Example - *stable closure* of $\mathcal{K}$:

\[
\text{st}(\mathcal{K}) = \mathcal{T} \cup \text{st}(\mathcal{T}) \cup \mathcal{P}
\]

\[
\forall x. \text{PERSON}(x) \rightarrow \text{AGENT}(x)
\]

\[
\text{AGENT}(\text{David})
\]

\[
\forall x. \text{PERSON}(x) \lor \neg \text{PERSON}(x)
\]

\[
\forall x. \text{AGENT}(x) \lor \neg \text{AGENT}(x)
\]

\[
\forall x. \text{PERSON}(x) \leftarrow \text{pcmember}(x, \text{LPNMR}) \land \text{AGENT}(x) \land \neg \text{machine}(x)
\]

\[
\text{pcmember}(\text{David}, \text{LPNMR})
\]

There is a classical model of this theory

\[
\mathcal{M} = \{\neg \text{PERSON}(\text{David}), \text{machine}(\text{David}), \ldots\}
\]

Thus:

\[
K \not\models_{\text{FOL}} \text{PERSON}(\text{David})
\]
Example - *stable closure* of \( \mathcal{K} \):

\[
\text{st}(\mathcal{K}) = T \cup \text{st}(T) \cup P
\]

\[
\forall x. \text{PERSON}(x) \rightarrow \text{AGENT}(x)
\]

\[
\text{AGENT}(\text{David})
\]

\[
\forall x. \text{PERSON}(x) \lor \neg \text{PERSON}(x)
\]

\[
\forall x. \text{AGENT}(x) \lor \neg \text{AGENT}(x)
\]

\[
\forall x. \text{PERSON}(x) \leftarrow \text{pcmember}(x, \text{LPNMR}) \land \text{AGENT}(x) \land \neg \text{machine}(x)
\]

\[
\text{pcmember}(\text{David}, \text{LPNMR})
\]

There IS a classical model of this theory

\[
\mathcal{M} = \{ \neg \text{PERSON}(\text{David}), \text{machine}(\text{David}), \ldots \}
\]

Thus:

\[
K \not\models_{\textit{FOL}} \text{PERSON}(\text{David})
\]
Example - *stable closure* of $\mathcal{K}$:

$$st(\mathcal{K}) = T \cup st(T) \cup \mathcal{P}$$

$$\forall x.\text{PERSON}(x) \rightarrow \text{AGENT}(x)$$
$$\text{AGENT}(\text{David})$$

$$\forall x.\text{PERSON}(x) \lor \neg \text{PERSON}(x)$$

$$\forall x.\text{AGENT}(x) \lor \neg \text{AGENT}(x)$$

$$\forall x.\text{PERSON}(x) \leftarrow \text{pcmember}(x, \text{LPNMR}), \text{AGENT}(x) \land \neg \text{machine}(x)$$

$$\text{pcmember}(\text{David}, \text{LPNMR})$$

The total HT-model $\mathcal{M}_{HT} = \langle H, T \rangle$ corresponding to $\mathcal{M}$ with:

$H = T = \{ \text{machine}(\text{David}), \ldots \}$

is NO Equilibrium model, since there is a model $\mathcal{M}'_{HT} \subset \mathcal{M}_{HT}$:

with: $H' = \{ \ldots \}$
$$T = \{ \text{machine}(\text{David}), \ldots \}$$

All Equilibrium models include $\text{PERSON}(\text{David})$, thus:

$$st(\mathcal{K}) \models_{QEL} \text{PERSON}(\text{David}) \checkmark$$
Example - *stable closure* of $\mathcal{K}$:

$$st(\mathcal{K}) = T \cup st(T) \cup P$$

\[
\begin{align*}
\forall x.\text{PERSON}(x) & \rightarrow \text{AGENT}(x) \\
\text{AGENT}(David) & \\
\forall x.\text{PERSON}(x) & \lor \neg \text{PERSON}(x) \\
\forall x.\text{AGENT}(x) & \lor \neg \text{AGENT}(x) \\
\forall x.\text{PERSON}(x) & \leftarrow \text{pcmember}(x, LPNMR), \text{AGENT}(x) \land \neg \text{machine}(x) \\
\text{pcmember}(David, LPNMR)
\end{align*}
\]

The total HT-model $\mathcal{M}_{HT} = \langle H, T \rangle$ corresponding to $\mathcal{M}$ with:

$H = T = \{\text{machine}(David), \ldots\}$

is NO Equilibrium model, since there is a model $\mathcal{M}'_{HT} \subsetneq \mathcal{M}_{HT}$:

with: $H' = \{\ldots\}$

$T = \{\text{machine}(David), \ldots\}$

All Equilibrium models include $\text{PERSON}(David)$, thus:

$$st(\mathcal{K}) \models_{QEL} \text{PERSON}(David) \checkmark$$
Example - stable closure of $\mathcal{K}$:

$$st(\mathcal{K}) = T \cup st(T) \cup P$$

- $\forall x.\text{PERSON}(x) \rightarrow \text{AGENT}(x)$
- $\text{AGENT}(\text{David})$
- $\forall x.\text{PERSON}(x) \lor \neg\text{PERSON}(x)$
- $\forall x.\text{AGENT}(x) \lor \neg\text{AGENT}(x)$
- $\forall x.\text{PERSON}(x) \leftarrow \text{pcmember}(x, \text{LPNMR}), \text{AGENT}(x) \land \neg\text{machine}(x)$
- $\text{pcmember}(\text{David}, \text{LPNMR})$

The total HT-model $\mathcal{M}_{HT} = \langle H, T \rangle$ corresponding to $\mathcal{M}$ with:

$H = T = \{\text{machine}(\text{David}), \ldots\}$

is NO Equilibrium model, since there is a model $\mathcal{M}'_{HT} \triangleleft \mathcal{M}_{HT}$:

with: $H' = \{\ldots\}$

$T = \{\text{machine}(\text{David}), \ldots\}$

All Equilibrium models include $\text{PERSON}(\text{David})$, thus:

$st(\mathcal{K}) \models_{QEL} \text{PERSON}(\text{David}) \checkmark$
Example - *stable closure* of $\mathcal{K}$:

$$st(\mathcal{K}) = T \cup st(T) \cup P$$

$$\forall x. PERSON(x) \rightarrow AGENT(x)$$

$$AGENT(David)$$

$$\forall x. PERSON(x) \lor \neg PERSON(x)$$

$$\forall x. AGENT(x) \lor \neg AGENT(x)$$

$$\forall x. PERSON(x) \leftarrow pcmember(x, LPNMR), AGENT(x) \land \neg machine(x)$$

$$pcmember(David, LPNMR)$$

The total HT-model $\mathcal{M}_{HT} = \langle H, T \rangle$ corresponding to $\mathcal{M}$ with:

$H = T = \{machine(David), \ldots\}$

is NO Equilibrium model, since there is a model $\mathcal{M}'_{HT} \triangleleft \mathcal{M}_{HT}$:

with: $H' = \{\ldots\}$

$T = \{machine(David), \ldots\}$

All Equilibrium models include $PERSON(David)$, thus:

$$st(\mathcal{K}) \models_{QEL} PERSON(David) \checkmark$$
Conclusions/Observations:

- Quantified Equilibrium Logic provides a powerful and intuitive tool as a “carrier” logic for Hybrid KBs.
- Embedding is simple: add LEM for classical predicates.
- Why this works is not so surprising: $QHT^s$ based on intuitionistic logic, adding LEM enforces totalization of HT models on the respective predicates, i.e. make them “classical”.
- No reducts/grounding involved, this gives us:
  - a semantics for nested logic programs. Well-investigated for propositional LPs, first-order case needs more investigation, respective results on QEL relatively new.
  - results on strong equivalence for Equilibrium Logic carry over: HT-model equivalence amounts to strong equivalence! Important notion for nonmonotonic logic programs, program optimization, meaning preserving transformations of hybrid KB’s, etc.
Conclusions/Observations:

- Quantified Equilibrium Logic provides a powerful and intuitive tool as a “carrier” logic for Hybrid KBs.
- Embedding is simple: add LEM for classical predicates.
- Why this works is not so surprising: $QHT^s$ based on intuitionistic logic, adding LEM enforces totalization of HT models on the respective predicates, i.e. make them “classical”.
- No reducts/grounding involved, this gives us:
  - a semantics for nested logic programs. Well-investigated for propositional LPs, first-order case needs more investigation, respective results on QEL relatively new.
  - results on strong equivalence for Equilibrium Logic carry over: HT-model equivalence amounts to strong equivalence!
    Important notion for nonmonotonic logic programs, program optimization, meaning preserving transformations of hybrid KB’s, etc.
Conclusions/Observations:

- Quantified Equilibrium Logic provides a powerful and intuitive tool as a “carrier” logic for Hybrid KBs.
- Embedding is simple: add LEM for classical predicates.
- Why this works is not so surprising: $QHT^s$ based on intuitionistic logic, adding LEM enforces totalization of HT models on the respective predicates, i.e. make them “classical”.
- No reducts/grounding involved, this gives us:
  - a semantics for nested logic programs. Well-investigated for propositional LPs, first-order case needs more investigation, respective results on QEL relatively new.
  - results on strong equivalence for Equilibrium Logic carry over: HT-model equivalence amounts to strong equivalence! Important notion for nonmonotonic logic programs, program optimization, meaning preserving transformations of hybrid KB’s, etc.
Conclusions/Observations:

- Quantified Equilibrium Logic provides a powerful and intuitive tool as a “carrier” logic for Hybrid KBs.
- Embedding is simple: add LEM for classical predicates.
- Why this works is not so surprising: $QHT_s$ based on intuitionistic logic, adding LEM enforces totalization of HT models on the respective predicates, i.e. make them “classical”.
- No reducts/grounding involved, this gives us:
  - a semantics for nested logic programs. Well-investigated for propositional LPs, first-order case needs more investigation, respective results on QEL relatively new.
  - results on strong equivalence for Equilibrium Logic carry over: HT-model equivalence amounts to strong equivalence! Important notion for nonmonotonic logic programs, program optimization, meaning preserving transformations of hybrid KB’s, etc.
Conclusions/Observations:

- Quantified Equilibrium Logic provides a powerful and intuitive tool as a “carrier” logic for Hybrid KBs.
- Embedding is simple: add LEM for classical predicates.
- Why this works is not so surprising: $QHT^s$ based on intuitionistic logic, adding LEM enforces totalization of HT models on the respective predicates, i.e. make them “classical”.
- No reducts/grounding involved, this gives us:
  - a semantics for nested logic programs. Well-investigated for propositional LPs, first-order case needs more investigation, respective results on QEL relatively new.
  - results on strong equivalence for Equilibrium Logic carry over: HT-model equivalence amounts to strong equivalence! Important notion for nonmonotonic logic programs, program optimization, meaning preserving transformations of hybrid KB’s, etc.
Conclusions/Observations:

- Quantified Equilibrium Logic provides a powerful and intuitive tool as a “carrier” logic for Hybrid KBs.
- Embedding is simple: add LEM for classical predicates.
- Why this works is not so surprising: $QHT^s$ based on intuitionistic logic, adding LEM enforces totalization of HT models on the respective predicates, i.e. make them “classical”.
- No reducts/grounding involved, this gives us:
  - a semantics for nested logic programs. Well-investigated for propositional LPs, first-order case needs more investigation, respective results on QEL relatively new.
  - results on strong equivalence for Equilibrium Logic carry over: HT-model equivalence amounts to strong equivalence!
  - Important notion for nonmonotonic logic programs, program optimization, meaning preserving transformations of hybrid KB’s, etc.
Conclusions/Observations:

- Quantified Equilibrium Logic provides a powerful and intuitive tool as a “carrier” logic for Hybrid KBs.
- Embedding is simple: add LEM for classical predicates.
- Why this works is not so surprising: $QHT^s$ based on intuitionistic logic, adding LEM enforces totalization of HT models on the respective predicates, i.e. make them “classical”.
- No reducts/grounding involved, this gives us:
  - a semantics for nested logic programs. Well-investigated for propositional LPs, first-order case needs more investigation, respective results on QEL relatively new.
  - results on strong equivalence for Equilibrium Logic carry over: HT-model equivalence amounts to strong equivalence! Important notion for nonmonotonic logic programs, program optimization, meaning preserving transformations of hybrid KB’s, etc.
Future Work:

- Paper covers only equality-free FOL embedding (results already there in [Pearce & Valverde, 2006])
- Investigation on related (IJCAI) approaches:
  - Logic of minimal knowledge and negation as failure (MKNF) [Motik & Rosati, 2007]
  - First-Order Autoepistemic Logic [de Bruijn et al., 2007]
  - Circumscription [Ferraris, Lee, Lifschitz, 2007]

Get the paper at: http://polleres.net/publications.html
Variants discussed in the paper:

<table>
<thead>
<tr>
<th>Variant</th>
<th>UNA</th>
<th>variables</th>
<th>disj.</th>
<th>neg. $\mathcal{L}_T$ atoms</th>
</tr>
</thead>
<tbody>
<tr>
<td>r-hybrid [Rosati, 2005]</td>
<td>yes</td>
<td>$\mathcal{L}_P$-safe</td>
<td>pos.</td>
<td>no</td>
</tr>
<tr>
<td>$r^+$-hybrid [Rosati, 2005b]</td>
<td>no</td>
<td>$\mathcal{L}_P$-safe</td>
<td>pos.</td>
<td>no</td>
</tr>
<tr>
<td>$r_w$-hybrid [Rosati, 2006]</td>
<td>yes</td>
<td>weak $\mathcal{L}_P$-safe</td>
<td>pos.</td>
<td>no</td>
</tr>
<tr>
<td>g-hybrid [Heymans, et al. 2006]</td>
<td>no</td>
<td>guarded</td>
<td>neg.*</td>
<td>yes</td>
</tr>
</tbody>
</table>

* g-hybrid allows negation in the head but at most one positive head atom

**Table:** Different variants of hybrid KBs