1. Answer Set Programs and HEX-Programs

**An Answer Set (ASP)-Program** is a set of rules of kind
\[
a_1 \vee \ldots \vee a_k \leftarrow b_1, \ldots, b_m, \neg b_{m+1}, \ldots, \neg b_n,
\]
where \( q_i \) for \( 1 \leq i \leq k \) and \( b_i \) for \( 1 \leq i \leq n \) are classical atoms. An answer set of such a program \( P \) is an interpretation \( I \) (a set of atoms), which is a subset-minimal model of the GL-reduct \( P^I \).

**HEX-programs** extend ASP by external sources:
- Rule bodies may contain external atoms of the form \( \varphi[q_1, \ldots, q_k](t_1, \ldots, t_l) \),
  
  where
  
  \( p \) : external predicate name, 
  
  \( q \) : predicate names or constants: \( \tau(\varphi, i) \in \{ \text{pred}, \text{const} \} \), 
  
  \( t \) : terms.

**Semantics:**
\[ 1 + k + \ell \text{-ary Boolean oracle function } f_{\varphi} \cdot \varphi[q_1, \ldots, q_k](t_1, \ldots, t_l) \text{ is true under assignment } A \]
if \( f_{\varphi}(A, q_1, \ldots, q_k, t_1, \ldots, t_l) = T \).

**Answer sets** are defined similarly as for ordinary ASP, but using the FLP-reduct \( P^F \) [Faber et al., 2011] instead of the GL-reduct \( P^I \).

**Example:** Set Partitioning

\[
P = \begin{cases} 
  d(a_1), \ldots, d(a_n), \\
r_1: p(X) \leftarrow d(X), \text{diff}[d, q](X) \\
r_2: q(X) \leftarrow d(X), \text{diff}[d, p](X) 
\end{cases}
\]

2. Motivation

**Equivalence of ASP-programs:**
- Deciding equivalence of ASP-programs under program extensions received attention in the past.
- Possible application: program transformations and optimizations.

**Existing equivalence notions:** programs \( P \) and \( Q \) are called
- strongly equivalent [Lifschitz et al., 2001]
  
  if \( P \cup R \) and \( Q \cup R \) have the same answer sets for any program \( R \); 
  
  uniformly equivalent [Eiter and Fink, 2003]
  
  if \( P \cup Q \) and \( Q \cup R \) have the same answer sets for any set of facts \( R \); 
  
  \( (\mathcal{H}, \mathcal{B}) \)-equivalent [ Wolpert, 2007] 
  
  if \( P \cup R \) and \( Q \cup R \) have the same answer sets for all programs \( R \in \mathcal{P}(\mathcal{H}, \mathcal{B}) \) whose head resp. body atoms come only from \( \mathcal{H} \) resp. \( \mathcal{B} \).

**Question 1:** How can we decide if two programs have the same answer set?

**Question 2:** What can we say about inconsistency of HEX-programs?

**Challenge:** The support for external atoms and the use of the FLP-instead of the GL-reduct make the extension non-trivial.

**Contributions:**
- A generalization of the notion of \( (\mathcal{H}, \mathcal{B}) \)-equivalence to HEX-programs, i.e., a formal criterion for deciding if two HEX-programs are \( (\mathcal{H}, \mathcal{B}) \)-equivalent.
- This subsumes strong and uniform equivalence.
- A related criterion for deciding inconsistency of a HEX-program.
- Notably, the notion is also applicable to special cases of HEX-programs, such as well-known ASP extensions, e.g., aggregates, DL-programs and constraint ASP.

3. The Equivalence Criterion

The following result is a generalization of the one by Wolpert:

**Definition**

Given sets \( \mathcal{H}, \mathcal{B} \) of atoms, a pair \( (X, Y) \) of interpretations is an \( (\mathcal{H}, \mathcal{B}) \)-model of a program \( P \) if

1. \( Y \models P \) and for each \( Y' \subseteq Y \) with \( Y' \models P \) we have \( Y' \models \mathcal{H} \cup \mathcal{B} \); and
2. if \( X \subseteq Y \) then there exists an \( X' \subseteq Y \) with \( X' \models \mathcal{H} \cup \mathcal{B} \) such that \( (X', Y) \) is \( \mathcal{H}, \mathcal{B} \)-maximal for \( P \).

We denote the set of all \( (\mathcal{H}, \mathcal{B}) \)-models of a program \( P \) by \( \sigma(\mathcal{H}, \mathcal{B})(P) \).

**Theorem (Equivalence of HEX-Programs)**

For sets \( \mathcal{H}, \mathcal{B} \) of atoms and HEX-programs \( P \) and \( Q \), we have

\[ P \equiv_Q (\mathcal{H}, \mathcal{B}) \] if and only if \( \sigma(\mathcal{H}, \mathcal{B})(P) = \sigma(\mathcal{H}, \mathcal{B})(Q) \).

**Proof Idea:** A technique for external source inlining [Redl, 2017] can be exploited to apply proof ideas by Wolpert.

4. The Inconsistency Criteria

We provide two criteria based on models of the redact and unfounded sets (UFS) [Faber, 2005], respectively. Let \( P \) be a HEX-program. Then:

**Theorem (Inconsistency of a Program based on Its Reduct)**

Program \( P \cup R \) is inconsistent for all \( R \in \mathcal{P}(\mathcal{H}, \mathcal{B}) \) iff for each model \( Y \) of \( P \) there is an \( Y' \subseteq Y \) such that \( Y' \models P^F \) and \( Y' \models \mathcal{H} \).

**Theorem (Inconsistency of a Program based on Unfounded Sets)**

Program \( P \cup R \) is inconsistent for all \( R \in \mathcal{P}(\mathcal{H}, \mathcal{B}) \) iff for each model \( Y \) of \( P \) there is a UFS \( U \neq \emptyset \) of \( P \) w.r.t. \( Y \) s.t. \( U \cap Y \neq \emptyset \) and \( U \cap \mathcal{H} = \emptyset \).

The latter theorem is especially useful for solver development since implementations do not usually explicitly construct the reduct.

5. Conclusion and Outlook

**Main results:**
- Decision criteria for
  1. equivalence and 
  2. inconsistency of HEX-programs.

**Future work:**
- Extension of the results to non-ground programs.
- Applications: program transformations for solver optimizations.

6. References