The Concept of a Stable Model:
   Early History
   and Some Recent Developments

Vladimir Lifschitz
University of Texas at Austin
1976: Maarten van Emden and Robert Kowalski, *The semantics of predicate logic as a programming language*.

1978: Keith Clark, *Negation as failure*.


1980: Raymond Reiter, *A logic for default reasoning*.

1982: Raymond Reiter, *Circumscription implies predicate completion (sometimes)*.

1985: Robert Moore, *Semantical considerations on nonmonotonic logic*. 

\[ r(X) \leftarrow p(X), \text{not } q(X). \]

\[ r(a) \leftarrow p(a), \text{not } q(a). \]

\[ r(b) \leftarrow p(b), \text{not } q(b). \]

\[ p(a) \land \neg L q(a) \rightarrow r(a), \]

\[ p(b) \land \neg L q(b) \rightarrow r(b). \]
1987: Nicole Bidoit and Christine Froidevaux, *Minimalism subsumes default logic and circumscription in stratified logic programming.*

\[
\begin{align*}
    r(X) & \leftarrow p(X), \text{not } q(X). \\
    p(X) : M \neg q(X) & \implies r(X).
\end{align*}
\]

Reduct relative to $M$:

(i) drop each rule $A_0 \leftarrow A_1, \ldots, A_m, not A_{m+1}, not A_n$ containing a term $not A_i$ with $A_i \in M$, and

(ii) drop all terms $not A_i$ from the remaining rules.


\[
\text{equilibrium logic} \quad \frac{\text{circumscription}}{\text{classical models}} = \quad \text{Kripke models}
\]

This characterization

- treats ground rules as shorthand for propositional formulas: 
  \[ r(a) \leftarrow p(a), \text{not} \ q(a) \text{ is identified with } p(a) \land \neg q(a) \rightarrow r(a); \]
- refers to Kripke models, but not to reducts;
- does not emphasize the role of negation: \( \neg A \) can be viewed as shorthand for \( A \rightarrow \bot \).
### ASP Constructs vs. Propositional formulas

<table>
<thead>
<tr>
<th>ASP Construct</th>
<th>Propositional formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q ; r ← p$</td>
<td>$p → q ∨ r$</td>
</tr>
<tr>
<td>${q, r} ← p$</td>
<td>$p → (q ∨ \neg q) ∧ (r ∨ \neg r)$</td>
</tr>
<tr>
<td>$← p, not q$</td>
<td>$\neg (p ∧ \neg q)$</td>
</tr>
<tr>
<td>$s ← 1{p, q, r}$</td>
<td>$p ∨ q ∨ r → s$</td>
</tr>
</tbody>
</table>
Traditional definition of the reduct:

(i) drop each rule $A_0 \leftarrow A_1, \ldots, A_m, \text{not } A_{m+1}, \text{not } A_n$ containing a term $\text{not } A_i$ with $A_i \in M$, and

(ii) drop all terms $\text{not } A_i$ from the remaining rules.


drop each rule $A_0 \leftarrow A_1, \ldots, A_m, \text{not } A_{m+1}, \text{not } A_n$
containing a term $A_i$ such that $A_i \notin M$ or a term $\text{not } A_i$
such $A_i \in M$. 

The reduct of a propositional formula $F$ relative to $M$ is obtained from $F$ by replacing every maximal subformula of $F$ that is not satisfied by $M$ with $\bot$.


Stable models of a first-order formula $F$ are defined as the models of $F$ that satisfy a certain condition expressed in second-order logic.

The semantics of RASPL-1 translates the rule

\[
\{ p(X) \} \leftarrow q(X), \{ Y : r(X,Y) \} \]

into the formula

\[
\forall X[q(X) \land \neg \exists Y_1 Y_2 (r(X,Y_1) \land q(X,Y_2) \land Y_1 \neq Y_2) \rightarrow p(X) \lor \neg p(X)].
\]
Conclusion

The invention of stable models was motivated by desire to “justify” the use of negation in logic programming.

Later on, this work led to the emergence of a valuable knowledge representation language.

From this perspective, the initial emphasis on the problem of negation appears to be a historical accident.