Syllogistic Reasoning
under the Weak Completion Semantics

Emmanuelle-Anna Dietz and Steffen Hölldobler

International Center for Computational Logic
TU Dresden, Germany

8th South-East Asian Summer School on Computational Logic

July 2016
Syllogisms

All B are A
All B are C

What follows?
Syllogisms

All B are A
All B are C

What follows?

All A are C  No A are C  Some A are C  Some A are not C
No C are A  Some C are A  Some C are not A  NVC
Reasoning Towards An Appropriate Logical Form

<table>
<thead>
<tr>
<th>Mood</th>
<th>NL</th>
<th>FOL</th>
<th>Short</th>
</tr>
</thead>
<tbody>
<tr>
<td>Affirmative universal (A)</td>
<td>All $a$ are $b$.</td>
<td>$\forall X (a(X) \rightarrow b(X))$</td>
<td>Aab</td>
</tr>
<tr>
<td>Affirmative existential (I)</td>
<td>Some $a$ are $b$.</td>
<td>$\exists X (a(X) \land b(X))$</td>
<td>lab</td>
</tr>
<tr>
<td>Negative universal (E)</td>
<td>No $a$ are $b$.</td>
<td>$\forall X (a(X) \rightarrow \neg b(X))$</td>
<td>Eab</td>
</tr>
<tr>
<td>Negative existential (O)</td>
<td>Some $a$ are not $b$.</td>
<td>$\exists X (a(X) \land \neg b(X))$</td>
<td>Oab</td>
</tr>
</tbody>
</table>
### Reasoning Towards An Appropriate Logical Form

<table>
<thead>
<tr>
<th>Mood</th>
<th>NL</th>
<th>FOL</th>
<th>Short</th>
</tr>
</thead>
<tbody>
<tr>
<td>Affirmative universal (A)</td>
<td>All a are b.</td>
<td>$\forall X (a(X) \rightarrow b(X))$</td>
<td>Aab</td>
</tr>
<tr>
<td>Affirmative existential (I)</td>
<td>Some a are b.</td>
<td>$\exists X (a(X) \land b(X))$</td>
<td>lab</td>
</tr>
<tr>
<td>Negative universal (E)</td>
<td>No a are b.</td>
<td>$\forall X (a(X) \rightarrow \neg b(X))$</td>
<td>Eab</td>
</tr>
<tr>
<td>Negative existential (O)</td>
<td>Some a are not b.</td>
<td>$\exists X (a(X) \land \neg b(X))$</td>
<td>Oab</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Figure 1</th>
<th>Figure 2</th>
<th>Figure 3</th>
<th>Figure 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Premise</td>
<td>a-b</td>
<td>b-a</td>
<td>a-b</td>
</tr>
<tr>
<td>Second Premise</td>
<td>b-c</td>
<td>c-b</td>
<td>c-b</td>
</tr>
</tbody>
</table>
Reasoning Towards An Appropriate Logical Form

<table>
<thead>
<tr>
<th>Mood</th>
<th>NL</th>
<th>FOL</th>
<th>Short</th>
</tr>
</thead>
<tbody>
<tr>
<td>Affirmative universal (A)</td>
<td>All $a$ are $b$.</td>
<td>$\forall X(a(X) \to b(X))$</td>
<td>Aab</td>
</tr>
<tr>
<td>Affirmative existential (I)</td>
<td>Some $a$ are $b$.</td>
<td>$\exists X(a(X) \land b(X))$</td>
<td>lab</td>
</tr>
<tr>
<td>Negative universal (E)</td>
<td>No $a$ are $b$.</td>
<td>$\forall X(a(X) \to \neg b(X))$</td>
<td>Eab</td>
</tr>
<tr>
<td>Negative existential (O)</td>
<td>Some $a$ are not $b$.</td>
<td>$\exists X(a(X) \land \neg b(X))$</td>
<td>Oab</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Figure 1</th>
<th>Figure 2</th>
<th>Figure 3</th>
<th>Figure 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Premise</td>
<td>a-b</td>
<td>b-a</td>
<td>a-b</td>
</tr>
<tr>
<td>Second Premise</td>
<td>b-c</td>
<td>c-b</td>
<td>c-b</td>
</tr>
</tbody>
</table>

- There are 64 different pairs of premises and 512 different pairs of syllogisms.
- A problem can be completely specified by the quantifiers of the first and second premise and the figure. The example just discussed is denoted by AA4.
Modeling Syllogisms

We model the Weak Completion Semantics to syllogisms and follow four principles:

1. Licenses for inferences
2. Existential Import and Gricean Implicature
3. Negation by Transformation
4. Unknown Generalization
According to Stenning and van Lambalgen [2008], conditionals should be formalized by licenses for inferences:

\[ p(X) \leftarrow q(X). \]

becomes

\[
\begin{align*}
p(X) & \leftarrow q(X) \land \neg ab(X). \\
ab(X) & \leftarrow \bot.
\end{align*}
\]
Humans normally do not quantify over things that do not exist.

Consequently, *for all* implies *there exists*.

Humans require existential import for a conditional to be true.
Logic programs do not allow negative literals as heads of clauses. Replace a negative conclusion $\neg p(X)$ by $p'(X)$ and add

$$
P(X) \leftarrow \neg p'(X).
U \leftarrow p(X) \land p'(X).
$$

where the second clause represents an integrity constraint.
Logic programs do not allow negative literals as heads of clauses. Replace a negative conclusion \( \neg p(X) \) by \( p'(X) \) and add

\[
\begin{align*}
p(X) & \leftarrow \neg p'(X). \\
U & \leftarrow p(X) \land p'(X).
\end{align*}
\]

where the second clause represents an integrity constraint.

Combined with the principle of licenses for inferences, we obtain

\[
\begin{align*}
p(X) & \leftarrow \neg p'(X) \land \neg ab(X). \\
ab(X) & \leftarrow \bot. \\
U & \leftarrow p(X) \land p'(X).
\end{align*}
\]
Humans seem to distinguish between some $a$ are $b$ and some $b$ are $a$.
But in FOL, $\exists X (a(X) \land b(X)) \equiv \exists X (b(X) \land a(X))$.
Humans seem to distinguish between some $a$ are $b$ and all $a$ are $b$.
If we learn that some $a$ are $b$, then
- there must be an object $o_1$ belonging to $a$ and $b$ (Gricean Implicature),
- there must be another object $o_2$ belonging to $a$ and for which it is unknown whether it belongs to $b$ (Unknown Generalization).
All $y$ are $z$

‘All $y$ are $z$’ is represented by the program $P_{Ay}$ which consists of the following clauses:

\[
\begin{align*}
z(X) & \leftarrow y(X) \land \neg ab_{yz}(X). \\
ab_{yz}(X) & \leftarrow \bot. \\
y(o) & \leftarrow \top.
\end{align*}
\]
All y are z

‘All y are z’ is represented by the program $\mathcal{P}_{Ayz}$ which consists of the following clauses:

\[
\begin{align*}
z(X) & \leftarrow y(X) \land \neg ab_{yz}(X). \\
ab_{yz}(X) & \leftarrow \bot. \\
y(o) & \leftarrow \top.
\end{align*}
\]

- The first two clauses are obtained by the principle of licenses for inferences.
All $y$ are $z$

‘All $y$ are $z$’ is represented by the program $\mathcal{P}_{Ayz}$ which consists of the following clauses:

\[
\begin{align*}
z(X) & \leftarrow y(X) \land \neg ab_{yz}(X). \\
ab_{yz}(X) & \leftarrow \bot. \\
y(o) & \leftarrow \top.
\end{align*}
\]

- The first two clauses are obtained by the principle of licenses for inferences.
- The last clause follows by the principle of Gricean implicature.
All y are z

‘All y are z’ is represented by the program $\mathcal{P}_{AyZ}$ which consists of the following clauses:

\[
\begin{align*}
z(X) & \leftarrow y(X) \land \neg ab_{yz}(X). \\
ab_{yz}(X) & \leftarrow \bot. \\
y(o) & \leftarrow \top.
\end{align*}
\]

- The first two clauses are obtained by the principle of licenses for inferences.
- The last clause follows by the principle of Gricean implicature.

The least model of the weak completion of $\mathcal{P}_{AyZ}$ is

\[
\langle \{y(o), z(o)\}, \{ab_{yz}(o)\} \rangle.
\]
‘No y are z’ in FOL can have different logical representations:

\[
\neg \exists X (y(X) \land z(X)) \\
\equiv \forall X \neg(y(X) \land z(X)) \quad \text{by} \quad \neg \exists X \equiv \forall \neg X, \\
\equiv \forall X (\neg y(X) \lor \neg z(X)) \quad \text{by} \quad \neg(A \land B) \equiv (\neg A \lor \neg B), \\
\equiv \forall X (\neg z(X) \lor \neg y(X)) \quad \text{by} \quad (A \lor B) \equiv (B \lor A), \\
\equiv \forall X (z(X) \rightarrow \neg y(X)) \quad \text{by} \quad (\neg A \lor B) \equiv (A \rightarrow B) \\
\equiv \forall X (y(X) \rightarrow \neg z(X)) \quad \text{by} \quad A \rightarrow B \equiv \neg A \lor B \\
\equiv \forall X (z(X) \rightarrow \neg y(X)) \quad \text{by} \quad A \rightarrow B \equiv \neg B \rightarrow \neg A \text{ and } \neg \neg A \equiv A
\]
No y are z (2)

\( P_{Ey} \) consists of the following clauses:

\[
\begin{align*}
\text{y}'(X) & \leftarrow z(X) \land \neg ab_{zny}(X). \\
ab_{zny}(X) & \leftarrow \bot. \\
y(X) & \leftarrow \neg \text{y}'(X) \land \neg ab_{nyy}(X). \\
z(o) & \leftarrow \top. \\
ab_{nyy}(o) & \leftarrow \bot.
\end{align*}
\]

In addition we have the following integrity constraint:

\[
U \leftarrow y(X) \land \text{y}'(X).
\]
No y are z (2)

$P_{Eyz}$ consists of the following clauses:

\[
\begin{align*}
y'(X) & \leftarrow z(X) \land \neg ab_{zny}(X). \\
ab_{zny}(X) & \leftarrow \bot. \\
y(X) & \leftarrow \neg y'(X) \land \neg ab_{nyy}(X). \\
z(o) & \leftarrow \top. \\
ab_{nyy}(o) & \leftarrow \bot.
\end{align*}
\]

In addition we have the following integrity constraint:

\[
U \leftarrow y(X) \land y'(X).
\]

- The first two clauses in $P_{Eyz}$ are obtained by licenses for inferences.
No y are z (2)

\[ P_{Eyz} \text{ consists of the following clauses:} \]

\begin{align*}
y'(X) & \leftarrow z(X) \land \neg ab_{zny}(X). \\
ab_{zny}(X) & \leftarrow \bot. \\
y(X) & \leftarrow \neg y'(X) \land \neg ab_{nyy}(X). \\
z(o) & \leftarrow \top. \\
ab_{nyy}(o) & \leftarrow \bot.
\end{align*}

In addition we have the following integrity constraint:

\[ U \leftarrow y(X) \land y'(X). \]

- The first two clauses in \( P_{Eyz} \) are obtained by licenses for inferences.
- The third clause applying the principle of negation by transformation.
No y are z (2)

$P_{Eyz}$ consists of the following clauses:

\[
\begin{align*}
y'(X) & \leftarrow z(X) \land \neg ab_{zny}(X). \\
ab_{zny}(X) & \leftarrow \bot. \\
y(X) & \leftarrow \neg y'(X) \land \neg ab_{nyy}(X). \\
z(o) & \leftarrow \top. \\
ab_{nyy}(o) & \leftarrow \bot.
\end{align*}
\]

In addition we have the following integrity constraint:

\[
U \leftarrow y(X) \land y'(X).
\]

- The first two clauses in $P_{Eyz}$ are obtained by licenses for inferences.
- The third clause applying the principle of negation by transformation.
- In addition, this principle enforces the integrity constraint.
No y are z (2)

$\mathcal{P}_{Eyz}$ consists of the following clauses:

\[
\begin{align*}
  y'(X) & \leftarrow z(X) \land \neg ab_{zny}(X). \\
  ab_{zny}(X) & \leftarrow \bot. \\
  y(X) & \leftarrow \neg y'(X) \land \neg ab_{nvy}(X). \\
  z(o) & \leftarrow \top. \\
  ab_{nvy}(o) & \leftarrow \bot.
\end{align*}
\]

In addition we have the following integrity constraint:

\[
U \leftarrow y(X) \land y'(X).
\]

- The first two clauses in $\mathcal{P}_{Eyz}$ are obtained by licenses for inferences.
- The third clause applying the principle of negation by transformation.
- In addition, this principle enforces the integrity constraint.
- The last two clauses of $\mathcal{P}_{Eyz}$ follows by the principle of Gricean implicature.
No y are z (2)

\( \mathcal{P}_{Ey} \) consists of the following clauses:

\[
\begin{align*}
y'(X) & \leftarrow z(X) \land \neg ab_{zny}(X). \\
ab_{zny}(X) & \leftarrow \bot. \\
y(X) & \leftarrow \neg y'(X) \land \neg ab_{nyy}(X). \\
z(o) & \leftarrow \top. \\
ab_{nyy}(o) & \leftarrow \bot.
\end{align*}
\]

In addition we have the following integrity constraint:

\[
U \leftarrow y(X) \land y'(X).
\]

- The first two clauses in \( \mathcal{P}_{Ey} \) are obtained by licenses for inferences.
- The third clause applying the principle of negation by transformation.
- In addition, this principle enforces the integrity constraint.
- The last two clauses of \( \mathcal{P}_{Ey} \) follows by the principle of Gricean implicature.

The least model of the weak completion of \( \mathcal{P}_{Ey} \) is

\[\langle \{z(o), y'(o)\}, \{ab_{zny}(o), ab_{nyy}(o), y(o)\} \rangle.\]
Some y are z

‘Some y are z’ represented by the $P_{lyz}$, which consists of the following clauses:

\[
\begin{align*}
z(X) & \leftarrow y(X) \land \neg ab_{yz}(X). \\
ab_{yz}(o_1) & \leftarrow \bot. \\
y(o_1) & \leftarrow \top. \\
y(o_2) & \leftarrow \top.
\end{align*}
\]
Some y are z

‘Some y are z’ represented by the $P_{lyz}$, which consists of the following clauses:

\[
\begin{align*}
z(X) & \leftarrow y(X) \land \neg ab_{yz}(X). \\
ab_{yz}(o_1) & \leftarrow \bot. \\
y(o_1) & \leftarrow \top. \\
y(o_2) & \leftarrow \top.
\end{align*}
\]

- The first two clauses are again obtained by the principle of using licenses for inferences.
Some $y$ are $z$

‘Some $y$ are $z$’ represented by the $\mathcal{P}_{lyz}$, which consists of the following clauses:

\[
\begin{align*}
  z(X) & \leftarrow y(X) \land \neg ab_{yz}(X). \\
  ab_{yz}(o_1) & \leftarrow \bot. \\
  y(o_1) & \leftarrow T. \\
  y(o_2) & \leftarrow T.
\end{align*}
\]

- The first two clauses are again obtained by the principle of using licenses for inferences.
- The abnormality predicate is restricted to the object $o_1$, which is assumed to exist by the principle of Gricean implicature, represented by the third clause.
Some y are z

‘Some y are z’ represented by the $\mathcal{P}_{lyz}$, which consists of the following clauses:

\[
\begin{align*}
z(X) & \leftarrow y(X) \land \neg abyz(X). \\
abyz(o_1) & \leftarrow \bot. \\
y(o_1) & \leftarrow \top. \\
y(o_2) & \leftarrow \top.
\end{align*}
\]

- The first two clauses are again obtained by the principle of using licenses for inferences.
- The abnormality predicate is restricted to the object $o_1$, which is assumed to exist by the principle of Gricean implicature, represented by the third clause.
- The fourth clause is obtained by the principle of unknown generalization.
Some y are z

‘Some y are z’ represented by the $\mathcal{P}_{lyz}$, which consists of the following clauses:

\[
\begin{align*}
z(X) & \leftarrow y(X) \land \neg ab_{yz}(X). \\
ab_{yz}(o_1) & \leftarrow \bot. \\
y(o_1) & \leftarrow \top. \\
y(o_2) & \leftarrow \top.
\end{align*}
\]

- The first two clauses are again obtained by the principle of using licenses for inferences.
- The abnormality predicate is restricted to the object $o_1$, which is assumed to exist by the principle of Gricean implicature, represented by the third clause.
- The fourth clause is obtained by the principle of unknown generalization.

The least model of the weak completion of $\mathcal{P}_{lyz}$ is

$$\langle \{y(o_1), y(o_2), z(o_1)\}, \{ab_{yz}(o_1)\} \rangle.$$
Some $y$ are not $z$

‘Some $y$ are not $z$’ represented by $P_{Oyz}$ consists of the following clauses:

\[
\begin{align*}
  z'(X) & \leftarrow y(X) \land \neg ab_{ynz}(X). \\
  ab_{ynz}(o_1) & \leftarrow \bot. \\
  z(X) & \leftarrow \neg z'(X) \land \neg ab_{nzz}(X). \\
  y(o_1) & \leftarrow \top. \\
  y(o_2) & \leftarrow \top. \\
  ab_{nzz}(o_1) & \leftarrow \bot. \\
  ab_{nzz}(o_2) & \leftarrow \bot.
\end{align*}
\]

In addition, we need the integrity constraint

\[
U \leftarrow z(X) \land z'(X).
\]
Some y are not z

‘Some y are not z’ represented by $\mathcal{P}_{Oyz}$ consists of the following clauses:

\begin{align*}
    z'(X) & \leftarrow y(X) \land \neg ab_{ynz}(X). \\
    ab_{ynz}(o_1) & \leftarrow \bot. \\
    z(X) & \leftarrow \neg z'(X) \land \neg ab_{nzz}(X). \\
    y(o_1) & \leftarrow \top. \\
    y(o_2) & \leftarrow \top. \\
    ab_{nzz}(o_1) & \leftarrow \bot. \\
    ab_{nzz}(o_2) & \leftarrow \bot.
\end{align*}

In addition, we need the integrity constraint

\[ U \leftarrow z(X) \land z'(X). \]

- The first four clauses and the integrity constraints are derived as in $\mathcal{P}_{Eyz}$. 
Some y are not z

‘Some y are not z’ represented by $\mathcal{P}_{Oyz}$ consists of the following clauses:

\[
\begin{align*}
z'(X) & \leftarrow y(X) \land \neg ab_{ynz}(X). \\
ab_{ynz}(o_1) & \leftarrow \bot. \\
z(X) & \leftarrow \neg z'(X) \land \neg ab_{nzz}(X). \\
y(o_1) & \leftarrow \top. \\
y(o_2) & \leftarrow \top. \\
ab_{nzz}(o_1) & \leftarrow \bot. \\
ab_{nzz}(o_2) & \leftarrow \bot.
\end{align*}
\]

In addition, we need the integrity constraint

\[
U \leftarrow z(X) \land z'(X).
\]

- The first four clauses and the integrity constraints are derived as in $\mathcal{P}_{Eyz}$.
- The fifth clause of $\mathcal{P}_{Oyz}$ is obtained by the principle of unknown generalization.
Some y are not z

'Some y are not z' represented by $\mathcal{P}_{Oyz}$ consists of the following clauses:

\[
\begin{align*}
z'(X) & \leftarrow y(X) \land \neg ab_{ynz}(X). \\
ab_{ynz}(o_1) & \leftarrow \bot. \\
z(X) & \leftarrow \neg z'(X) \land \neg ab_{nzz}(X). \\
y(o_1) & \leftarrow \top. \\
y(o_2) & \leftarrow \top. \\
ab_{nzz}(o_1) & \leftarrow \bot. \\
ab_{nzz}(o_2) & \leftarrow \bot.
\end{align*}
\]

In addition, we need the integrity constraint

\[
U \leftarrow z(X) \land z'(X).
\]

- The first four clauses and the integrity constraints are derived as in $\mathcal{P}_{Eyz}$.
- The fifth clause of $\mathcal{P}_{Oyz}$ is obtained by the principle of unknown generalization.

The least model of the weak completion of $\mathcal{P}_{Oyz}$ is

\[
\langle \{y(o_1), y(o_2), z'(o_1)\}, \{ab_{ynz}(o_1), ab_{nzz}(o_1), ab_{nzz}(o_2), z(o_1)\} \rangle.
\]
Three Examples
Syllogism AA4

All B are A
All B are C

What follows?
Syllogism AA4

All B are A
All B are C

What follows?

All A are C  No A are C  Some A are C  Some A are not C
All C are A  No C are A  Some C are A  Some C are not A  NVC
Syllogism AA4

All B are A
All B are C

What follows?

All A are C  No A are C  Some A are C  Some A are not C
All C are A  No C are A  Some C are A  Some C are not A  NVC

Valid Conclusions
All B are A
All B are C

What follows?

All A are C  No A are C  Some A are C  Some A are not C
All C are A  No C are A  Some C are A  Some C are not A  NVC

Majority's Conclusions
Syllogism AA4

\( \mathcal{P}_{AA4} \) consists of the following clauses:

\[
\begin{align*}
  a(X) & \leftarrow b(X) \land \neg ab_{ba}(X) \\
  b(o_1) & \leftarrow \top \\
  ab_{ba}(X) & \leftarrow \bot \\
  c(X) & \leftarrow b(X) \land \neg ab_{bc}(X) \\
  ab_{bc}(X) & \leftarrow \bot \\
  b(o_2) & \leftarrow \top
\end{align*}
\]

The least model of the weak completion of \( \mathcal{P}_{AA4} \) is

\[
\langle \{b(o_1), b(o_2), a(o_1), a(o_2), c(o_1), c(o_2)\},
\{ab_{ba}(o_1), ab_{ba}(o_2), ab_{bc}(o_1), ab_{bc}(o_2)\}\rangle.
\]

- This model entails both ‘all a are c’ and ‘all c are a’.
- Analogously this also holds for ‘all c are a’.
- This prediction matches partially with the answers from participants who concluded Aac and NVC.
Some b are not a
All b are c

What follows?
Some b are not a
All b are c

What follows?

All A are C No A are C Some A are C Some A are not C
All C are A No C are A Some C are A Some C are not A NVC
Syllogism OA4

**Some b are not a**

**All b are c**

What follows?

- All A are C
- No A are C
- Some A are C
- Some A are not C

- All C are A
- No C are A
- Some C are A
- Some C are not A

**Valid Conclusion**
Syllogism OA4

Some B are not A
All B are C

What follows?

All A are C  No A are C  Some A are C  Some A are not C
All C are A  No C are A  Some C are A  Some C are not A  NVC

Majority’s Conclusions
No B are A
All C are B

What follows?
No B are A
All C are B

What follows?

All A are C  No A are C  Some A are C  Some A are not C
All C are A  No C are A  Some C are A  Some C are not A  NVC
Syllogism EA2

No b are a
All c are b

What follows?

All A are C  No A are C  Some A are C  Some A are not C
All C are A  No C are A  Some C are A  Some C are not A  NVC

Valid Conclusions
Syllogism EA2

No B are A
All C are B

What follows?

All A are C  No A are C  Some A are C  Some A are not C
All C are A  No C are A  Some C are A  Some C are not A  NVC

Majority’s Conclusions