Declarative Problem Solving through Abduction

Antonis C. Kakas
Department of Computer Science
University of Cyprus
CYPRUS (CHYPRE)
antonis@ucy.ac.cy
(Subject Title: “DPS-Paris2007”)

PART 4

11-18 January, 2007

Paris, France
Course Breakdown

- Introduction
- Abduction – General Introduction
- Modelling Problems for Abduction and DPS
- Computational Logic & PROLOG – Background
- Abductive Logic Programming – Semantics
- **Abductive Logic Programming** – Computation
- ALP for Declarative Problem Solving – Diagnosis
- Projects – Discussion

cnt...
Lecture 4: Computing Abduction in ALP

- **Aim:** To study in detail the computational model of ALP. To be aware of the proof procedures for computing abductive explanations in ALP in the presence of NAF and integrity constraints.
Computing Abductive Explanations in (ALP)

- **Extend SLD to ASLD, ASLDIC, ASLDICNF**

- We will assume abducible predicates are not defined in $P$ at all. If some abducible predicate, $h$, has a partial definition in $P$ then:

  - Let $h$ if $B$ be a clause in $P$. Let also $h'$ be a new atom, not occurring in $P$, and let $P'$ be $P \cup \{h$ if $h'\}$ and $H'$ be $H - \{h\} \cup \{h'\}$.

  - Then $<P', H'>$ is equivalent to $<P, H>$, in that for every $O$, there exists an explanation $E$ for $O$ wrt $<P, H>$ iff there exist an explanation $E'$ for $O$ wrt $<P', H'>$.
Computing Abductive Explanations in (ALP)

- Extend SLD
- Assume abducible predicates are not defined in P at all.
- If some abducible predicate, h, has a partial definition in P then:
  - Let h if B be a clause in T
  - Let h' be a new atom, not occurring in T, and let
  - T' be T ∪ {h if h'} and H' be H - {h} ∪ {h'}.

- Then <T', H'> is equivalent to <T, H>, in that for every O, there exists an explanation E for O wrt <T, H> iff there exist an explanation E' for O wrt <T', H'>.
SLD-resolution - RECAP

- See supplement slides on SLD
A SLD derivation is a sequence:
\[<(G_0, \theta_0), (G_1, \theta_1), \ldots, (G_{n-1}, \theta_{n-1}), (\emptyset, \emptyset)>\]

Given \((G_i, \theta_i), 0 \leq i < n\), the next element, \((G_{i+1}, \theta_{i+1})\), is given by:

- If \( \exists \) resolvent \( G \):
  - \( G_{i+1} = G_i \)
  - \( \text{R selects } A \text{ in } G_i \)
  - \( \text{Success} \)

- Otherwise:
  - \( \text{Fail} \)

\( y \)
\( n \)
Given an abductive logic program \( <T,H> \), an abductive SLD-derivation for an observation \( O \) (wrt a safe selection rule \( R \)) consists of

- a sequence of goals \(<G_0=O, G_1, \ldots>\),
- a sequence of mgus \(<\theta_0=\{\}, \theta_1, \theta_2, \ldots>\),
- a sequence of (variants of) rules in \( T < C_0=\{\}, C_1, C_2, \ldots> \)
- a sequence of sets of hypotheses \(<E_0=\{\}, E_1, E_2, \ldots>\)

such that, if \( A \) is selected by \( R \) in \( G_i \), then:

1. If \( A \in H \) and \( A \in E_i \),
   \[ G_{i+1}=G_i-\{A\}, \quad E_{i+1}=E_i, \quad \theta_{i+1}=\{\}, \quad C_{i+1}=A, \]
2. If \( A \in H \) but \( A \notin E_i \),
   \[ G_{i+1}=G_i-\{A\}, \quad E_{i+1}=E_i\cup\{A\}, \quad \theta_{i+1}=\{\}, \quad C_{i+1}=A, \]
3. If \( A \notin H \), then \( E_{i+1}=E_i \) and
   \[ G_{i+1} \text{ is the resolvent of } G_i \text{ and } C_{i+1} \text{ via } \theta_{i+1} \text{ on the atom selected by } R \text{ in } G_i \text{ and the head of } C_{i+1}. \]
Abductive SLD-resolution (2)

- A safe selection rule selects any atom $A$ of an abducible predicate only if $A$ is ground.

- Abductive SLD-refutation/successful derivation/finitely failed derivation/tree.

- We represent an abductive refutation in short as $\langle (O\text{bs}, \theta_0, E_0), \ldots, (G_{n-1}, \theta_{n-1}, E_{n-1}), (\ldots, \theta_n, E_n) \rangle$ with $\theta_0 = \{\}, E_0 = \{\}$

- We say that the refutation:
  - computes the answer $\theta$ obtained by restricting the composition of $\theta_1, \ldots, \theta_n$ to the variables of $O\text{bs}$
  - computes the set of hypotheses $E_n$ for $O\text{bs}$.
Example (abductive SLD-refutations)

- **T:** wobbly-wheel(X) if broken-spokes(X)
  wobbly-wheel(X) if flat-tyre(X)
  flat-tyre(X) if leaky-valve(X)
  flat-tyre(X) if punctured-tube(X)

- **H:** broken-spokes(b), leaky-valve(b), punctured-tube(b)

- **O:** wobbly-wheel(b)

- `<(?ww(b),{},{}), (?bs(b),{},{}),( ,{},{}),{bs(b)})`
- `<(?ww(b),{},{}), (?ft(b),{},{}), (?lv(b),{},{}),( ,{},{}),{lv(b)})`
- `<(?ww(b),{},{}), (?ft(b),{},{}), (?pt(b),{},{}),( ,{},{}),{pt(b)})`

- E1={bs(b)}
- E2={lv(b)}
- E3={pt(b)}
ASLD at a glance

- \(<(\emptyset, \emptyset, E_0), \ldots, (G_{n-1}, \emptyset, E_{n-1}), (\emptyset, \emptyset, E_n)\rangle\) with \(\emptyset = \emptyset, E_0 = \emptyset\)
- Suppose we have built a derivation till \((G_i, \emptyset, E_i), 0 \leq i < n\).
- Then \((G_{i+1}, \emptyset, E_{i+1})\) is constructed as follows:

```plaintext
R selects A in Gi

A in H

A in Ei

\exists resolvent G

G_{i+1} = Gi, \{A\}
E_{i+1} = E_{i} \cup \{A\}

Gi = Gi - \{A\}

Ei = Ei

n = 1

Success

Fail

Gi+1 = Gi
Ei+1 = Ei
```
Computation of Explanations in the Presence of Integrity Constraints

- Given \(<T,H,IC>\) and \(O\), an explanation \(E\) (for \(O\)) can be computed via interleaving:
  - \textit{backward reasoning} (e.g. via ASLD) with \(T\) to generate \(E\)
  - \textit{forward reasoning} with \(T\) and \(IC\) to test \(E\)

- e.g. \(T=\{p \; \text{if} \; q, \; q \; \text{if} \; r, \; p \; \text{if} \; s\}\)
  \(H=\{r, s\}\)
  \(IC=\{\text{if} \; q \; \text{then} \; \text{false}\}\)

- \(O=p\)
  \(T \xleftarrow{p} \)
  \(T \xleftarrow{?q} \)
  \(T \xleftarrow{?r} \)
  \(T \xleftarrow{\text{false}} \)
  \(E=\{s\}\)
ASLD with testing of IC (ASLDIC)

- Assume IC is a set of denials, of the form
- **if** $A_1$ **and** ... **and** $A_m$ **then false**, $A_i$ atom, $m>0$
- with at least one $A_i$ an atom of an *abducible predicate*^{*}

- *abductive* (ASLD) derivations/refutations to *prove* observations/sentences (condition 1))
  $<(O, \{\}), \ldots ,(G_{n-1}, E_{n-1}), (\ldots , E_n)>$
- *consistency* derivations/refutations to *disprove* (residues of) integrity constraints (condition 2’))
  $<(S, E_0), \ldots ,(S_{n-1},E_{n-1}), (\{\}, E_n)>$

- *Abducible predicates* are predicates of hypotheses in $H$

- Note: we ignore mgus for simplicity
- Note: $E_i$ may contain *negations* of hypotheses
Given an abductive logic program \(<T,H,IC>\), an abductive derivation/refutation for an observation \(O\) (wrt \(R\)) is
\[
<(O, \{\}), \ldots , (G_n, E_n)> / \langle(O, \{\}), \ldots , (\cdot, E_n)\rangle
\]
such that, if \(A\) is selected by \(R\) in \(G_i\), \(0 \leq i < n\), then:

1. If \(A \in H\) and \(A \in E_i\), then \(G_{i+1} = G_i - \{A\}\), \(E_{i+1} = E_i\),
2. If \(A \in H\) and \(A \not\in E_i\) and not \(A \not\in E_i\) let \(S\) be the set of all resolvents of \(A\) with \(IC\); if \(\cdot\) is not in \(S\) and there exists a consistency refutation \(<(S, E_i \cup \{A\}), \ldots , (\cdot, E)>\) then \(G_{i+1} = G_i - \{A\}\), \(E_{i+1} = E\),
3. If \(A \not\in H\), then \(E_{i+1} = E_i\) and \(G_{i+1}\) is the resolvent of \(G_i\) and a rule/fact \(C\) in \(T\) on the atom selected by \(R\) in \(G_i\) and the head of \(C\)

An abductive refutation \(<(O, \{\}), \ldots , (\cdot, E_n)>\) is said to compute the set of hypotheses \(E_n \cap H\) (and an answer \(\theta\))
Consistency derivation/refutation

- Given an abductive logic program \(<T,H,IC>\), a consistency derivation/refutation (wrt R) is

\[\langle(S_0,E_0), \ldots ,(S_n,E_n)\rangle / \langle(S_0,E_0),\ldots ,\{\},E_n)\rangle\]

such that, if G is selected in \(S_i\), \(0 \leq i < n\), and A is selected by R in G, then

1. If \(A \in H\) and \(A \in E_i\), then, if \(G \setminus \{A\} \neq \emptyset\), \(S_{i+1} = S_i \setminus \{G\} \cup \{G \setminus \{A\}\}\), \(E_{i+1} = E_i\)
2. If \(A \in H\), \(A \notin E_i\) and \(\text{not } A \notin E_i\), then \(S_{i+1} = S_i \setminus \{G\}\), \(E_{i+1} = E_i\)
3. If \(A \in H\), \(A \notin E_i\) and \(\text{not } A \notin E_i\), then \(S_{i+1} = S_i \setminus \{G\}\), \(E_{i+1} = E_i \cup \{\text{not } A\}\)
4. If \(A \notin H\), then let S be the set of all resolvents of G and a rule/fact C in T on the atom A selected by R in G and the head of C. Then, if \(\notin S\), \(S_{i+1} = S_i \setminus \{G\} \cup S\), \(E_{i+1} = E_i\)
Example (ASLDIC)

- **T:** wobbly-wheel(X) if broken-spokes(X)
  wobbly-wheel(X) if flat-tyre(X)
  flat-tyre(X) if leaky-valve(X)
  flat-tyre(X) if punctured-tube(X)
- **H:** broken-spokes(b), leaky-valve(b), punctured-tube(b)
- **IC:** if leaky-valve then false
- **O:** wobbly-wheel(b)

- `<(WW(b),{}),(BS(b),{}),(LV(b),{}),(PT(b),{}),(BS(b))>` refutation
- `<(WW(b),{}),(FT(b),{}),(LV(b),{}),(PT(b),{}),(BS(b))>` finitely failed
- `<(WW(b),{}),(FT(b),{}),(PT(b),{}),(BS(b))>` refutation
Example

- **T:** \( p(X) \text{ if } q(X) \text{ and } b(X) \)
- \( q(X) \text{ if } a(X) \)
- \( q(X) \text{ if } c(X) \)
- **IC:** \( \text{if } a(X) \text{ and } b(X) \text{ then false} \)
- **H:** all ground atoms of the predicates a,b,c
- **O:** \( p(3) \)

\(<\(?p(3),\{}\),(?q(3) \text{ and } b(3),\{}\),(?a(3) \text{ and } b(3),\{}\),(?b(3),\{a(3),\text{not } b(3)\})>)\)

- since \(<\(<\{?b(3)\},\{a(3)\}\),\{\},\{a(3),\text{not } b(3)\}>)\>

\(<\(?p(3),\{}\),(?q(3) \text{ and } b(3),\{}\),(?c(3) \text{ and } b(3),\{}\),
(?b(3),\{c(3)\}),\{\},\{c(3),b(3),\text{not } a(3)\})>)\>

- since \(<\(<\{?a(3)\},\{c(3),b(3)\}\),\{\},\{c(3),b(3),\text{not } a(3)\}>)\>

- Thus \{c(3),b(3)\} is the computed set of hypotheses
Graphical conventions

- **T**: p \textbf{if} a
- **H**: a, b, c
- **IC**: \textbf{if} a \textbf{and} b \textbf{then} false
  \quad \textbf{if} a \textbf{and} c \textbf{then} false
- **O**: p

- **R**: left-most selection rule

\[ E := \{\} \]

- \(?p\)
- \(?a\) \quad E := \{a\}
- \(?b\) \quad ?c

\[ E := \{a, \text{not } b\} \quad E := \{a, \text{not } b, \text{not } c\} \]

\[ E := \{a, \text{not } b, \text{not } c\} \]

\[ \]
ASLD derivation (at a glance)

- ASLD derivation: \(<(O, \{\}), \ldots, (G_{n-1}, E_{n-1}), (G_n, E_n)>\) where \((G_{i+1}, E_{i+1})\), is given by:

Let \(S\) be the set of all resolvents of \(A\) and IC

\[G_{i+1} = G_i \setminus \{A\}\]
\[E_{i+1} = E_i\]

\( \exists \) resolvent \(G\)

- \(G_i+1 = G_i\) \(\setminus \{A\}\)
- \(E_{i+1} = E_i\)

\(R\) selects \(A\) in \(G_i\)

- \(n = i\)

- \(G_i = \ldots\)

- \(\exists\) a consistency refutation from \((S, E_i \cup \{A\})\) to \((\{\}, E)\)

\(\exists\) a consistency refutation from \((S, E_i \cup \{A\})\) to \((\{\}, E)\)

\(\exists\) a consistency refutation from \((S, E_i \cup \{A\})\) to \((\{\}, E)\)
Consistency derivation (at a glance)

- A consistency derivation is \( <(S_0, E_0), \ldots, (S_{n-1}, E_{n-1}), (\{\}, E_n)> \)
- Given \((S_i, E_i), 0 \leq i < n\), we construct \((S_{i+1}, E_{i+1})\) as follows:

1. Select \( G \) in \( S_i \); \( R \) selects \( A \) in \( G \)
2. If \( A \) in \( H \), then \( S_{i+1} = S_i - \{G\} \cup \{G - \{A\}\} \)
3. If \( \text{not } A \) in \( E_i \), then \( S_{i+1} = S_i - \{G\}, E_{i+1} = E_i \)
4. If \( S \) all resolvents of \( G \) in \( S \), then \( S_{i+1} = S_i - \{G\} \cup S \), \( E_{i+1} = E_i \)
5. If \( S_i = \{\} \) and \( n = i \), then \( S_{i+1} = S_i - \{G\} \), \( E_{i+1} = E_i \)
Soundness and (lack of) completeness of ASLDIC

- Given abductive logic program \(<T,H,IC>\) and observation \(O\):
  - if there exists an abductive refutation for \(O\) (wrt some selection rule \(R\)) computing the answer \(\theta\) and the set of hypotheses \(E\), then \(E\) is an explanation for \(O\theta\) (*soundness of ASLDIC-resolution*)
  - Counterexample to *completeness of ASLDIC-resolution*:
    - \(T\): \(q\) \textit{if } \(a\) \hspace{1cm} \(H\): \(a\) \hspace{1cm} \(IC\): \textit{if } \(a\) \textit{and } \(p\) \textit{then false}\hspace{1cm} \(O\): \(q\)
    - \(p\) \textit{if } \(p\)
      - \(E=\{a\}\) (minimal) explanation for \(O\)
        - \(\text{since } LHM(T\cup\{a\})=\{a,q\}, \text{ and IC true in } \{a,q\}\)
      - \(E\) is not computed by ASLDIC
Minimality of computed explanations

Given abductive logic theory \(<T,H>\) and observation \(O\):

- if there exists an ASLD-refutation for \(O\) (wrt some selection rule \(R\)) computing the answer \(\theta\) and the set of hypotheses \(E\), then \(E\) is a **locally minimal explanation** for \(O\theta\), i.e. \(E\) is minimal wrt \(<T',H>\), for some \(T' \subseteq T\)

- e.g. \(T\): \(p\) if \(a\) \hspace{1cm} \(H\): \(a,b\) \hspace{1cm} \(O\): \(p\)

  \[p\] if \(a\) \hspace{1cm} \[p\] if \(a\) and \(b\)

  \(E_1=\{a\}\) and \(E_2=\{a,b\}\) both computed by abductive SLD
  \(E_1\) is minimal
  \(E_2\) is not, but it is minimal wrt \(T'\): \(p\) if \(a\) and \(b\)

- Similarly for ASLDIC and \(<T,H,IC>\) theories.

- Explanations computed by ASLD(IC) are **basic** (since abducible predicates are assumed to have no definition in the LP, \(T\).)
ASLDIC with negation as failure

Negation as Failure through abduction

- NAF literals are treated as positive abducibles and not \( p \) holds if the abductive hypothesis "not \( p \)" is consistent-satisfies a set of integrity constraints.

- A normal logic program \( P \) is transformed into an abductive framework \( <P^*, A^*, IC^*> \) where:
  - \( A^* = \{p^* \mid p^* \text{ new predicate symbol for each } p \in P\} \)
  - \( P^* \) is obtained from \( P \) by replacing each NAF condition "not \( p \)" by a positive abducible condition \( p^* \).
  - \( IC^* \) contains two constraints:
    
    \[
    \forall x. \neg(p(x) \land p^*(x)) \\
    \forall x. p(x) \lor p^*(x), \forall \text{ each } p \text{ in } P.
    \]
Negation as Failure through abduction

A normal logic program \( P \) is transformed into an abductive framework \( \langle P^*, A^*, IC^* \rangle \)

**Result:** NAF as abduction corresponds to stable models semantics

**Example**

\[
\begin{align*}
P &: \ p \leftarrow \text{not } q \\
q &: \ \text{not } r
\end{align*}
\]

\[
\begin{align*}
\Delta &= \{r^*, p^*\} \text{ satisfies } I^* \\
M &= \{q\} \text{ stable model of } P \\
\Delta &= \{q^*\} \text{ violates } I^*
\end{align*}
\]
Example (ASLDICN)

- **T:** $p \text{ if } q \text{ and not } r$
  - $q$
  - $r \text{ if not } p$
- **H=*/{} **IC=*/{} **O:** $p$
- **E=*/{} $\in E!$$E=${not r}

Computed explanation is {}
Example (ASLDICN)

- **T**: \( p \text{ if } q \text{ and not } r \)
- \( r \text{ if not } p \)

- **IC**: \( \text{if } q \text{ and } s \text{ then false} \)
- **H**: \( q,s \)  
  **Obs**: \( p \)

- **E**: \( \{q\} \)
- \( \{q, \text{not } r\} \)
- \( \{q, \text{not } r, \text{not } s\} \)