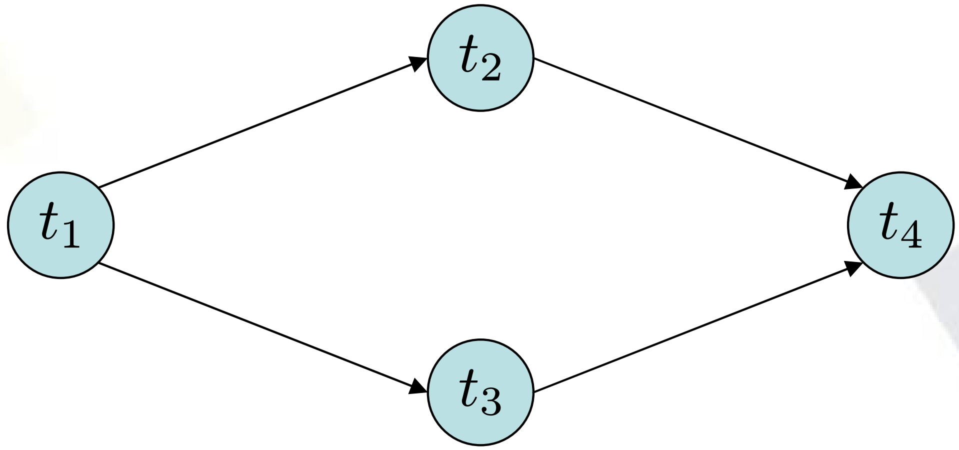
Learning Dependency Property from Traces

(CS294-2, Spring 2006)

Thomas Huining Feng
tfeng@eecs.berkeley.edu

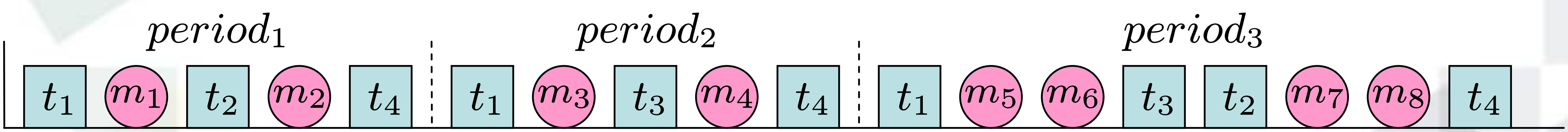
Apr. 27, 2006

Our Learning Problem



In this example, t_1 is a disjunction node, and t_4 is a conjunction node.

- ◇ A *disjunction node* is a node that can choose execution paths.
- ◇ A *conjunction node* is a node that passively receives messages from multiple execution paths.



Learn *node dependency* from traces, but

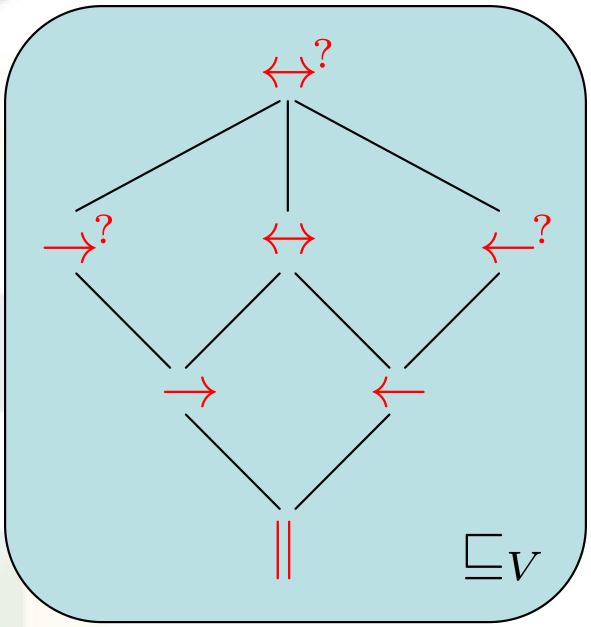
- ◇ Whether a node is disjunction/conjunction is unknown.
- ◇ Sender/receiver/meaning of a message (transmitted on the bus) is unknown.

Problem Definition (1)

Tasks. T is the set of tasks in the system (provided by the supplier).

Dependency function. $d \in D : T \times T \to V$, where

$$V = \{\|, \rightarrow, \leftarrow, \leftrightarrow, \rightarrow^?, \leftarrow^?, \leftrightarrow^?\}$$



- $\|$: t_1 and t_2 are in *parallel*.
- $\rightarrow$: if t_1 executes in a period, it *always* determines t_2.
- $\leftarrow$: if t_1 executes in a period, it *always* depends on t_2.
- $\leftrightarrow$: t_1 and t_2 *always* depend on each other.
- $\rightarrow^?$: if t_1 executes in a period, it *may or may not* determine t_2.
- $\leftarrow^?$: if t_1 executes in a period, it *may or may not* depend on t_2.
- $\leftrightarrow^?$: t_1 and t_2 *may or may not* depend on each other.

Problem Definition (2)

More-specific-than. $\sqsubseteq_V$ was defined as a lattice. $\sqsubseteq_D$ is the *pointwise order* of $\sqsubseteq_V$ (also a lattice). $\forall d_1, d_2 \in D$

$$d_1 \sqsubseteq_D d_2 \;\Leftrightarrow\; \forall t_1, t_2 \in T.(d_1(t_1, t_2) \sqsubseteq_V d_2(t_1, t_2))$$

The abstracted learning problem. Given

- I (the set of instances),
- $\langle D, \sqsubseteq_D \rangle$ (the space of possible dependency functions), and
- $M : D \times I \to boolean$, or $M : D \times \mathcal{P}(I) \to boolean$ (the matching function),

find $D^* \subseteq D$ s.t.

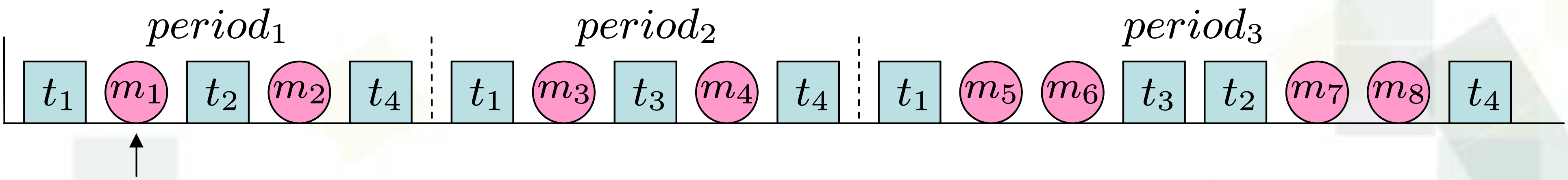
- *Correctness*:

$$\forall d^* \in D^*.M(d^*, I)$$

- *Completeness* and *optimality*:

$$\forall d \in D.(M(d, I) \Rightarrow \exists d^* \in D^*.d^* \sqsubseteq_D d)$$

A Simple Example of the Algorithm



d_0	t_1	t_2	t_3	t_4
t_1	$\|$	$\|$	$\|$	$\|$
t_2	$\|$	$\|$	$\|$	$\|$
t_3	$\|$	$\|$	$\|$	$\|$
t_4	$\|$	$\|$	$\|$	$\|$

$\Rightarrow$

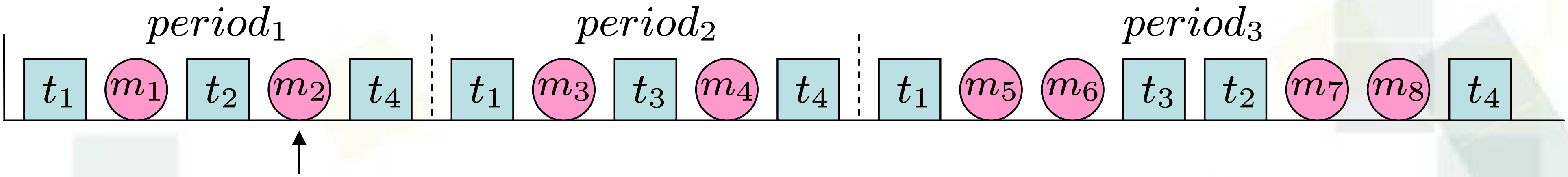
d_{11}	t_1	t_2	t_3	t_4
t_1	$\|$	$\rightarrow$	$\|$	$\|$
t_2	$\leftarrow$	$\|$	$\|$	$\|$
t_3	$\|$	$\|$	$\|$	$\|$
t_4	$\|$	$\|$	$\|$	$\|$

$m_1 : t_1 \mapsto t_2$

d_{12}	t_1	t_2	t_3	t_4
t_1	$\|$	$\|$	$\|$	$\rightarrow$
t_2	$\|$	$\|$	$\|$	$\|$
t_3	$\|$	$\|$	$\|$	$\|$
t_4	$\leftarrow$	$\|$	$\|$	$\|$

$m_1 : t_1 \mapsto t_4$

A Simple Example of the Algorithm



d_{11}	t_1	t_2	t_3	t_4
t_1	$\|$	$\rightarrow$	$\|$	$\|$
t_2	$\leftarrow$	$\|$	$\|$	$\|$
t_3	$\|$	$\|$	$\|$	$\|$
t_4	$\|$	$\|$	$\|$	$\|$

$m_1 : t_1 \mapsto t_2$

$\Rightarrow$

d_{21}	t_1	t_2	t_3	t_4
t_1	$\|$	$\rightarrow$	$\|$	$\rightarrow$
t_2	$\leftarrow$	$\|$	$\|$	$\|$
t_3	$\|$	$\|$	$\|$	$\|$
t_4	$\leftarrow$	$\|$	$\|$	$\|$

$m_1 : t_1 \mapsto t_2; m_2 : t_1 \mapsto t_4$

d_{22}	t_1	t_2	t_3	t_4
t_1	$\|$	$\rightarrow$	$\|$	$\|$
t_2	$\leftarrow$	$\|$	$\|$	$\rightarrow$
t_3	$\|$	$\|$	$\|$	$\|$
t_4	$\|$	$\leftarrow$	$\|$	$\|$

$m_1 : t_1 \mapsto t_2; m_2 : t_2 \mapsto t_4$

d_{12}	t_1	t_2	t_3	t_4
t_1	$\|$	$\|$	$\|$	$\rightarrow$
t_2	$\|$	$\|$	$\|$	$\|$
t_3	$\|$	$\|$	$\|$	$\|$
t_4	$\leftarrow$	$\|$	$\|$	$\|$

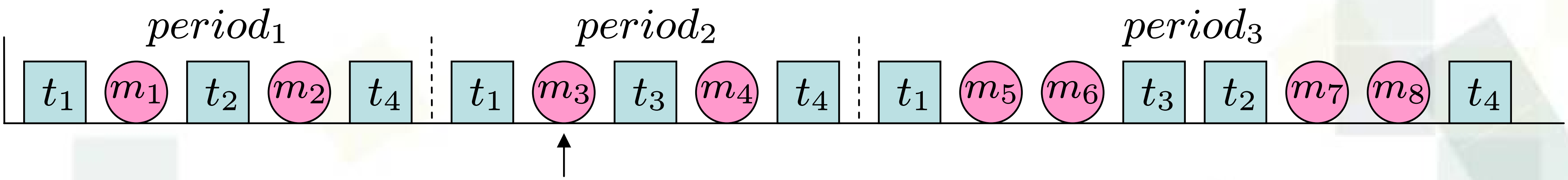
$m_1 : t_1 \mapsto t_4$

$\Rightarrow$

d_{23}	t_1	t_2	t_3	t_4
t_1	$\|$	$\|$	$\|$	$\rightarrow$
t_2	$\|$	$\|$	$\|$	$\rightarrow$
t_3	$\|$	$\|$	$\|$	$\|$
t_4	$\leftarrow$	$\leftarrow$	$\|$	$\|$

$m_1 : t_1 \mapsto t_4; m_2 : t_2 \mapsto t_4$

A Simple Example of the Algorithm



d_{21}	t_1	t_2	t_3	t_4
t_1	$\parallel$	$\rightarrow$	$\parallel$	$\rightarrow$
t_2	$\leftarrow$	$\parallel$	$\parallel$	$\parallel$
t_3	$\parallel$	$\parallel$	$\parallel$	$\parallel$
t_4	$\leftarrow$	$\parallel$	$\parallel$	$\parallel$

$\Rightarrow$

d_{31}	t_1	t_2	t_3	t_4
t_1	$\parallel$	$\rightarrow$	$\rightarrow$	$\rightarrow$
t_2	$\leftarrow$	$\parallel$	$\parallel$	$\parallel$
t_3	$\leftarrow$	$\parallel$	$\parallel$	$\parallel$
t_4	$\leftarrow$	$\parallel$	$\parallel$	$\parallel$

$m_3 : t_1 \mapsto t_3$

d_{31}	t_1	t_2	t_3	t_4
t_1	$\parallel$	$\rightarrow$	$\parallel$	$\rightarrow$
t_2	$\leftarrow$	$\parallel$	$\parallel$	$\parallel$
t_3	$\parallel$	$\parallel$	$\parallel$	$\parallel$
t_4	$\leftarrow$	$\parallel$	$\parallel$	$\parallel$

$m_3 : t_1 \mapsto t_4$

d_{22}	t_1	t_2	t_3	t_4
t_1	$\parallel$	$\rightarrow$	$\parallel$	$\parallel$
t_2	$\leftarrow$	$\parallel$	$\parallel$	$\rightarrow$
t_3	$\parallel$	$\parallel$	$\parallel$	$\parallel$
t_4	$\parallel$	$\leftarrow$	$\parallel$	$\parallel$

$\Rightarrow$

d_{22}	t_1	t_2	t_3	t_4
t_1	$\parallel$	$\rightarrow$	$\rightarrow$	$\parallel$
t_2	$\leftarrow$	$\parallel$	$\parallel$	$\rightarrow$
t_3	$\leftarrow$	$\parallel$	$\parallel$	$\parallel$
t_4	$\parallel$	$\leftarrow$	$\parallel$	$\parallel$

$m_3 : t_1 \mapsto t_3$

d_{22}	t_1	t_2	t_3	t_4
t_1	$\parallel$	$\rightarrow$	$\parallel$	$\rightarrow$
t_2	$\leftarrow$	$\parallel$	$\parallel$	$\rightarrow$
t_3	$\parallel$	$\parallel$	$\parallel$	$\parallel$
t_4	$\leftarrow$	$\leftarrow$	$\parallel$	$\parallel$

$m_3 : t_1 \mapsto t_4$

d_{23}	t_1	t_2	t_3	t_4
t_1	$\parallel$	$\parallel$	$\parallel$	$\rightarrow$
t_2	$\parallel$	$\parallel$	$\parallel$	$\rightarrow$
t_3	$\parallel$	$\parallel$	$\parallel$	$\parallel$
t_4	$\leftarrow$	$\leftarrow$	$\parallel$	$\parallel$

$\Rightarrow$

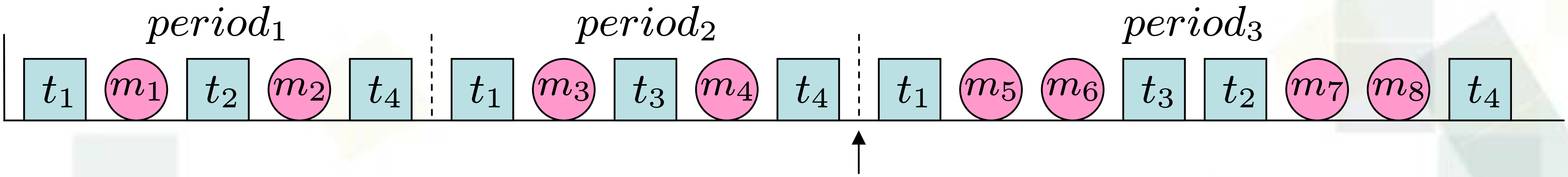
d_{23}	t_1	t_2	t_3	t_4
t_1	$\parallel$	$\parallel$	$\rightarrow$	$\rightarrow$
t_2	$\parallel$	$\parallel$	$\parallel$	$\rightarrow$
t_3	$\leftarrow$	$\parallel$	$\parallel$	$\parallel$
t_4	$\leftarrow$	$\leftarrow$	$\parallel$	$\parallel$

$m_3 : t_1 \mapsto t_3$

d_{23}	t_1	t_2	t_3	t_4
t_1	$\parallel$	$\parallel$	$\parallel$	$\rightarrow$
t_2	$\parallel$	$\parallel$	$\parallel$	$\rightarrow$
t_3	$\parallel$	$\parallel$	$\parallel$	$\parallel$
t_4	$\leftarrow$	$\leftarrow$	$\parallel$	$\parallel$

$m_3 : t_1 \mapsto t_4$

A Simple Example of the Algorithm



d_{41}	t_1	t_2	t_3	t_4
t_1	$\parallel$	$\rightarrow$	$\parallel$	$\rightarrow$
t_2	$\leftarrow$	$\parallel$	$\parallel$	$\parallel$
t_3	$\parallel$	$\parallel$	$\parallel$	$\rightarrow$
t_4	$\leftarrow$	$\parallel$	$\leftarrow$	$\parallel$

d_{42}	t_1	t_2	t_3	t_4
t_1	$\parallel$	$\parallel$	$\parallel$	$\rightarrow$
t_2	$\parallel$	$\parallel$	$\parallel$	$\rightarrow$
t_3	$\parallel$	$\parallel$	$\parallel$	$\rightarrow$
t_4	$\leftarrow$	$\leftarrow$	$\leftarrow$	$\parallel$

d_{43}	t_1	t_2	t_3	t_4
t_1	$\parallel$	$\rightarrow$	$\rightarrow$	$\rightarrow$
t_2	$\leftarrow$	$\parallel$	$\parallel$	$\parallel$
t_3	$\leftarrow$	$\parallel$	$\parallel$	$\parallel$
t_4	$\leftarrow$	$\parallel$	$\parallel$	$\parallel$

d_{44}	t_1	t_2	t_3	t_4
t_1	$\parallel$	$\parallel$	$\rightarrow$	$\rightarrow$
t_2	$\parallel$	$\parallel$	$\parallel$	$\rightarrow$
t_3	$\leftarrow$	$\parallel$	$\parallel$	$\parallel$
t_4	$\leftarrow$	$\leftarrow$	$\parallel$	$\parallel$

d_{45}	t_1	t_2	t_3	t_4
t_1	$\parallel$	$\rightarrow$	$\rightarrow$	$\rightarrow$
t_2	$\leftarrow$	$\parallel$	$\parallel$	$\rightarrow$
t_3	$\leftarrow$	$\parallel$	$\parallel$	$\parallel$
t_4	$\leftarrow$	$\leftarrow$	$\parallel$	$\parallel$

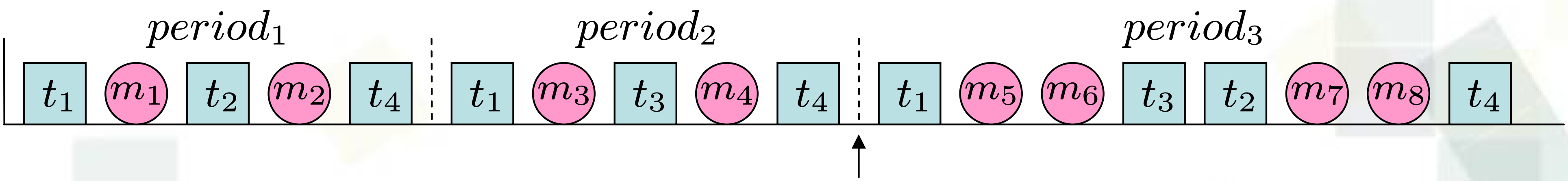
d_{46}	t_1	t_2	t_3	t_4
t_1	$\parallel$	$\rightarrow$	$\rightarrow$	$\rightarrow$
t_2	$\leftarrow$	$\parallel$	$\parallel$	$\parallel$
t_3	$\leftarrow$	$\parallel$	$\parallel$	$\rightarrow$
t_4	$\leftarrow$	$\parallel$	$\leftarrow$	$\parallel$

d_{47}	t_1	t_2	t_3	t_4
t_1	$\parallel$	$\rightarrow$	$\rightarrow$	$\parallel$
t_2	$\leftarrow$	$\parallel$	$\parallel$	$\rightarrow$
t_3	$\leftarrow$	$\parallel$	$\parallel$	$\rightarrow$
t_4	$\parallel$	$\leftarrow$	$\leftarrow$	$\parallel$

d_{48}	t_1	t_2	t_3	t_4
t_1	$\parallel$	$\parallel$	$\rightarrow$	$\rightarrow$
t_2	$\parallel$	$\parallel$	$\parallel$	$\rightarrow$
t_3	$\leftarrow$	$\parallel$	$\parallel$	$\rightarrow$
t_4	$\leftarrow$	$\leftarrow$	$\leftarrow$	$\parallel$

d_{49}	t_1	t_2	t_3	t_4
t_1	$\parallel$	$\rightarrow$	$\parallel$	$\rightarrow$
t_2	$\leftarrow$	$\parallel$	$\parallel$	$\rightarrow$
t_3	$\parallel$	$\parallel$	$\parallel$	$\rightarrow$
t_4	$\leftarrow$	$\leftarrow$	$\leftarrow$	$\parallel$

A Simple Example of the Algorithm



Post-processing 1: test conditional dependencies

d_{41}	t_1	t_2	t_3	t_4
t_1	$\Vert$	$\rightarrow^?$	$\Vert$	$\rightarrow$
t_2	$\leftarrow$	$\Vert$	$\Vert$	$\Vert$
t_3	$\Vert$	$\Vert$	$\Vert$	$\rightarrow$
t_4	$\leftarrow$	$\Vert$	$\leftarrow^?$	$\Vert$

d_{42}	t_1	t_2	t_3	t_4
t_1	$\Vert$	$\Vert$	$\Vert$	$\rightarrow$
t_2	$\Vert$	$\Vert$	$\Vert$	$\rightarrow$
t_3	$\Vert$	$\Vert$	$\Vert$	$\rightarrow$
t_4	$\leftarrow$	$\leftarrow^?$	$\leftarrow^?$	$\Vert$

d_{43}	t_1	t_2	t_3	t_4
t_1	$\Vert$	$\rightarrow^?$	$\rightarrow^?$	$\rightarrow$
t_2	$\leftarrow$	$\Vert$	$\Vert$	$\Vert$
t_3	$\leftarrow$	$\Vert$	$\Vert$	$\Vert$
t_4	$\leftarrow$	$\Vert$	$\Vert$	$\Vert$

d_{44}	t_1	t_2	t_3	t_4
t_1	$\Vert$	$\Vert$	$\rightarrow^?$	$\rightarrow$
t_2	$\Vert$	$\Vert$	$\Vert$	$\rightarrow$
t_3	$\leftarrow$	$\Vert$	$\Vert$	$\Vert$
t_4	$\leftarrow$	$\leftarrow^?$	$\Vert$	$\Vert$

d_{45}	t_1	t_2	t_3	t_4
t_1	$\Vert$	$\rightarrow^?$	$\rightarrow^?$	$\rightarrow$
t_2	$\leftarrow$	$\Vert$	$\Vert$	$\rightarrow$
t_3	$\leftarrow$	$\Vert$	$\Vert$	$\Vert$
t_4	$\leftarrow$	$\leftarrow^?$	$\Vert$	$\Vert$

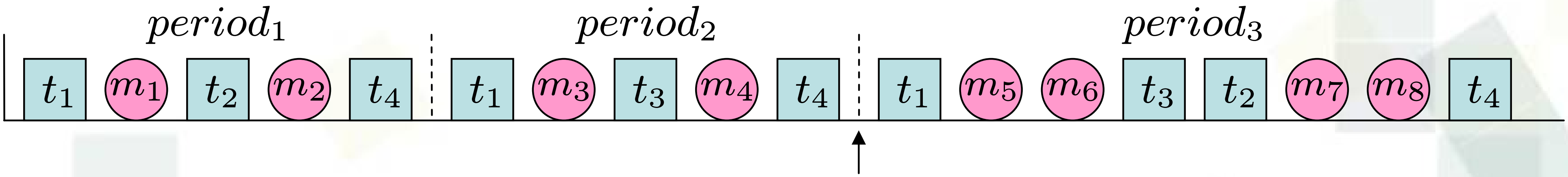
d_{46}	t_1	t_2	t_3	t_4
t_1	$\Vert$	$\rightarrow^?$	$\rightarrow^?$	$\rightarrow$
t_2	$\leftarrow$	$\Vert$	$\Vert$	$\Vert$
t_3	$\leftarrow$	$\Vert$	$\Vert$	$\rightarrow$
t_4	$\leftarrow$	$\Vert$	$\leftarrow^?$	$\Vert$

d_{47}	t_1	t_2	t_3	t_4
t_1	$\Vert$	$\rightarrow^?$	$\rightarrow^?$	$\Vert$
t_2	$\leftarrow$	$\Vert$	$\Vert$	$\rightarrow$
t_3	$\leftarrow$	$\Vert$	$\Vert$	$\rightarrow$
t_4	$\Vert$	$\leftarrow^?$	$\leftarrow^?$	$\Vert$

d_{48}	t_1	t_2	t_3	t_4
t_1	$\Vert$	$\Vert$	$\rightarrow^?$	$\rightarrow$
t_2	$\Vert$	$\Vert$	$\Vert$	$\rightarrow$
t_3	$\leftarrow$	$\Vert$	$\Vert$	$\rightarrow$
t_4	$\leftarrow$	$\leftarrow^?$	$\leftarrow^?$	$\Vert$

d_{49}	t_1	t_2	t_3	t_4
t_1	$\Vert$	$\rightarrow^?$	$\Vert$	$\rightarrow$
t_2	$\leftarrow$	$\Vert$	$\Vert$	$\rightarrow$
t_3	$\Vert$	$\Vert$	$\Vert$	$\rightarrow$
t_4	$\leftarrow$	$\leftarrow^?$	$\leftarrow^?$	$\Vert$

A Simple Example of the Algorithm



Post-processing 2: find redundant dependencies

d_{41}	t_1	t_2	t_3	t_4
t_1	$\parallel$	$\rightarrow^?$	$\parallel$	$\rightarrow$
t_2	$\leftarrow$	$\parallel$	$\parallel$	$\parallel$
t_3	$\parallel$	$\parallel$	$\parallel$	$\rightarrow$
t_4	$\leftarrow$	$\parallel$	$\leftarrow^?$	$\parallel$

d_{42}	t_1	t_2	t_3	t_4
t_1	$\parallel$	$\parallel$	$\parallel$	$\rightarrow$
t_2	$\parallel$	$\parallel$	$\parallel$	$\rightarrow$
t_3	$\parallel$	$\parallel$	$\parallel$	$\rightarrow$
t_4	$\leftarrow$	$\leftarrow^?$	$\leftarrow^?$	$\parallel$

d_{43}	t_1	t_2	t_3	t_4
t_1	$\parallel$	$\rightarrow^?$	$\rightarrow^?$	$\rightarrow$
t_2	$\leftarrow$	$\parallel$	$\parallel$	$\parallel$
t_3	$\leftarrow$	$\parallel$	$\parallel$	$\parallel$
t_4	$\leftarrow$	$\parallel$	$\parallel$	$\parallel$

d_{44}	t_1	t_2	t_3	t_4
t_1	$\parallel$	$\parallel$	$\rightarrow^?$	$\rightarrow$
t_2	$\parallel$	$\parallel$	$\parallel$	$\rightarrow$
t_3	$\leftarrow$	$\parallel$	$\parallel$	$\parallel$
t_4	$\leftarrow$	$\leftarrow^?$	$\parallel$	$\parallel$

d_{45}	t_1	t_2	t_3	t_4
t_1	$\parallel$	$\rightarrow^?$	$\rightarrow^?$	$\rightarrow$
t_2	$\leftarrow$	$\parallel$	$\parallel$	$\rightarrow$
t_3	$\leftarrow$	$\parallel$	$\parallel$	$\parallel$
t_4	$\leftarrow$	$\leftarrow^?$	$\parallel$	$\parallel$

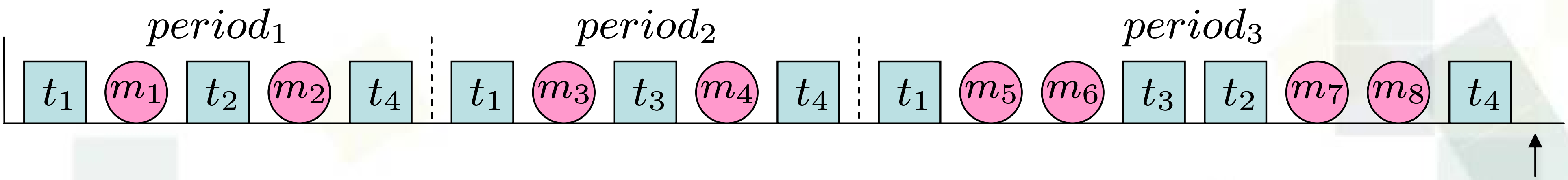
d_{46}	t_1	t_2	t_3	t_4
t_1	$\parallel$	$\rightarrow^?$	$\rightarrow^?$	$\rightarrow$
t_2	$\leftarrow$	$\parallel$	$\parallel$	$\parallel$
t_3	$\leftarrow$	$\parallel$	$\parallel$	$\rightarrow$
t_4	$\leftarrow$	$\parallel$	$\leftarrow^?$	$\parallel$

d_{47}	t_1	t_2	t_3	t_4
t_1	$\parallel$	$\rightarrow^?$	$\rightarrow^?$	$\parallel$
t_2	$\leftarrow$	$\parallel$	$\parallel$	$\rightarrow$
t_3	$\leftarrow$	$\parallel$	$\parallel$	$\rightarrow$
t_4	$\parallel$	$\leftarrow^?$	$\leftarrow^?$	$\parallel$

d_{48}	t_1	t_2	t_3	t_4
t_1	$\parallel$	$\parallel$	$\rightarrow^?$	$\rightarrow$
t_2	$\parallel$	$\parallel$	$\parallel$	$\rightarrow$
t_3	$\leftarrow$	$\parallel$	$\parallel$	$\rightarrow$
t_4	$\leftarrow$	$\leftarrow^?$	$\leftarrow^?$	$\parallel$

d_{49}	t_1	t_2	t_3	t_4
t_1	$\parallel$	$\rightarrow^?$	$\parallel$	$\rightarrow$
t_2	$\leftarrow$	$\parallel$	$\parallel$	$\rightarrow$
t_3	$\parallel$	$\parallel$	$\parallel$	$\rightarrow$
t_4	$\leftarrow$	$\leftarrow^?$	$\leftarrow^?$	$\parallel$

A Simple Example of the Algorithm



d_{41}	t_1	t_2	t_3	t_4
t_1	$\parallel$	$\rightarrow^?$	$\rightarrow^?$	$\rightarrow$
t_2	$\leftarrow$	$\parallel$	$\parallel$	$\parallel$
t_3	$\leftarrow$	$\parallel$	$\parallel$	$\rightarrow$
t_4	$\leftarrow$	$\parallel$	$\leftarrow^?$	$\parallel$

d_{42}	t_1	t_2	t_3	t_4
t_1	$\parallel$	$\parallel$	$\rightarrow^?$	$\rightarrow$
t_2	$\parallel$	$\parallel$	$\parallel$	$\rightarrow$
t_3	$\leftarrow$	$\parallel$	$\parallel$	$\rightarrow$
t_4	$\leftarrow$	$\leftarrow^?$	$\leftarrow^?$	$\parallel$

d_{43}	t_1	t_2	t_3	t_4
t_1	$\parallel$	$\rightarrow^?$	$\parallel$	$\rightarrow$
t_2	$\leftarrow$	$\parallel$	$\parallel$	$\rightarrow$
t_3	$\parallel$	$\parallel$	$\parallel$	$\rightarrow$
t_4	$\leftarrow$	$\leftarrow^?$	$\leftarrow^?$	$\parallel$

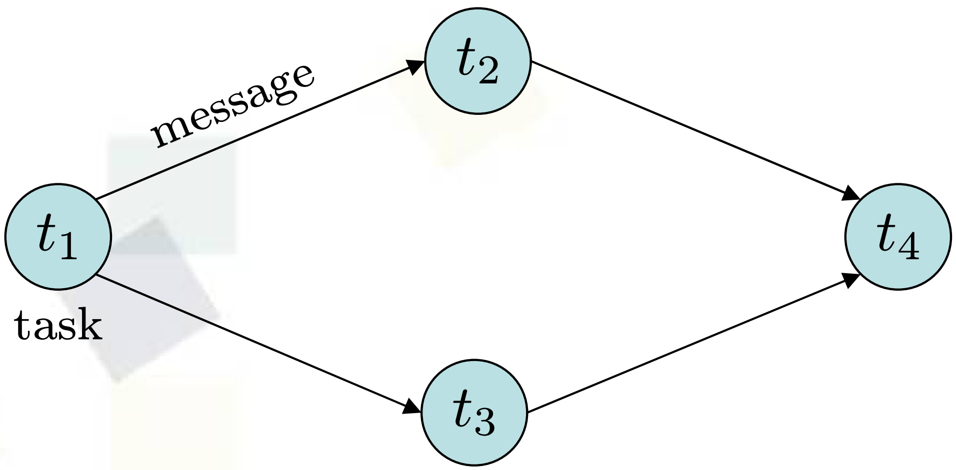
d_{44}	t_1	t_2	t_3	t_4
t_1	$\parallel$	$\rightarrow^?$	$\rightarrow^?$	$\rightarrow$
t_2	$\leftarrow$	$\parallel$	$\parallel$	$\rightarrow$
t_3	$\leftarrow$	$\parallel$	$\parallel$	$\parallel$
t_4	$\leftarrow$	$\leftarrow^?$	$\parallel$	$\parallel$

d_{45}	t_1	t_2	t_3	t_4
t_1	$\parallel$	$\rightarrow^?$	$\rightarrow^?$	$\parallel$
t_2	$\leftarrow$	$\parallel$	$\parallel$	$\rightarrow$
t_3	$\leftarrow$	$\parallel$	$\parallel$	$\rightarrow$
t_4	$\parallel$	$\leftarrow^?$	$\leftarrow^?$	$\parallel$

When the trace is finished and the algorithm does not converge, take the LUB (optional).

d_n	t_1	t_2	t_3	t_4
t_1	$\parallel$	$\rightarrow^?$	$\rightarrow^?$	$\rightarrow$
t_2	$\leftarrow$	$\parallel$	$\parallel$	$\rightarrow$
t_3	$\leftarrow$	$\parallel$	$\parallel$	$\rightarrow$
t_4	$\leftarrow$	$\leftarrow^?$	$\leftarrow^?$	$\parallel$

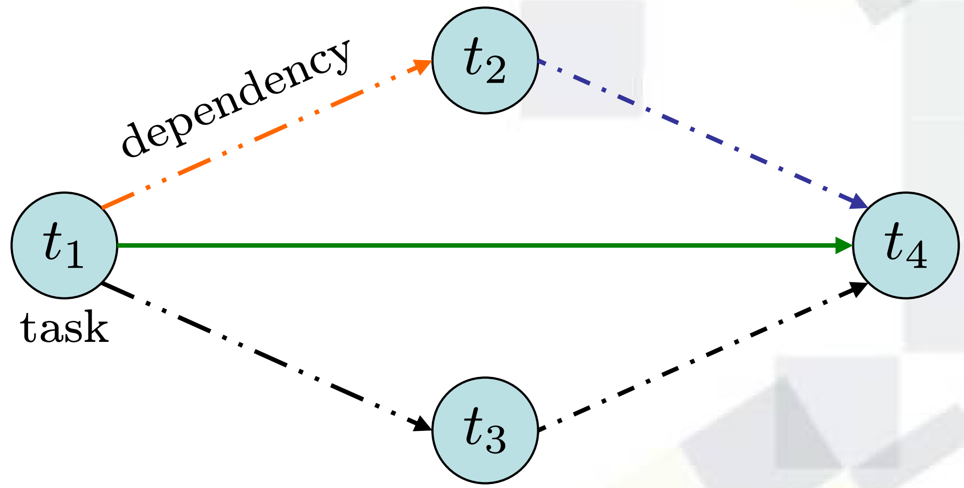
Construct Dependency Graph from the Result



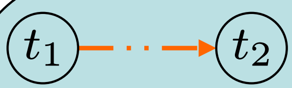
Original model

d_n	t_1	t_2	t_3	t_4
t_1	$\parallel$	$\rightarrow^?$	$\rightarrow^?$	$\rightarrow$
t_2	$\leftarrow$	$\parallel$	$\parallel$	$\rightarrow$
t_3	$\leftarrow$	$\parallel$	$\parallel$	$\rightarrow$
t_4	$\leftarrow$	$\leftarrow^?$	$\leftarrow^?$	$\parallel$

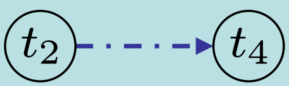
$\Downarrow$



Dependency graph



$d(t_1, t_2) = \rightarrow^? \ \wedge \ d(t_2, t_1) = \leftarrow$



$d(t_2, t_4) = \rightarrow \ \wedge \ d(t_4, t_2) = \leftarrow^?$

$d(t_1, t_4) = \rightarrow \ \wedge \ d(t_4, t_1) = \leftarrow$

t_1 is disjunction; t_4 is conjunction.

Properties of the Algorithm

Theorem 1 (NP-hard). The problem of finding the set of most specific hypotheses for a given trace is NP-hard.

Theorem 2 (Correctness). [a] The algorithm (with or without heuristics) guarantees correctness. I.e., if I is the set of instances in the trace, and D^* is the set of hypotheses that the algorithm returns, then $\forall d^* \in D^*. \forall i \in I. M(d^*, i)$.

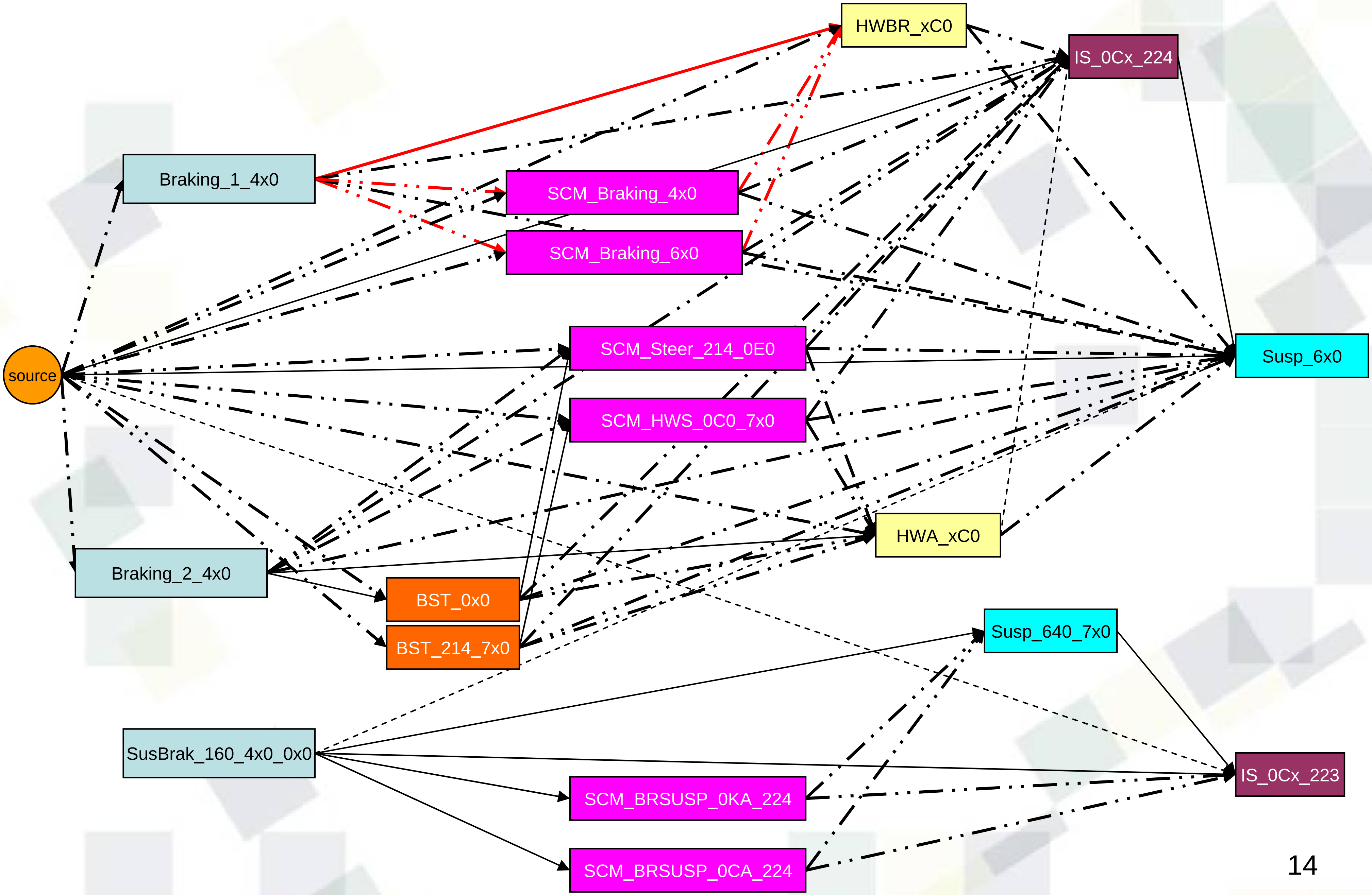
Theorem 3 (Optimality). The algorithm without heuristics guarantees optimality. I.e., if I is the set of instances in the trace, and D^* is the set of hypotheses that the algorithm returns, then $\forall d^* \in D^*. \overline{\exists} d' \in D. d' \sqsubset_D d^* \wedge \forall i \in I. M(d', i)$.

Lemma. If the algorithm returns the set of hypotheses D^* with the bound set to b, and d^* is the hypothesis obtained with the bound set to 1, then $d^* = D^*$ (the least upper bound of all the elements in D^*).

Theorem 4 (Convergence). If the algorithm converges to one hypothesis d_1^*, regardless of whether the bound is set or what the bound is, and if the algorithm returns d_2^* with the bound set to 1, then $d_1^* = d_2^*$.

[a] Heuristics is not discussed here.

An Industrial-Size Example



References

[1] Haibo Zeng, Abhijit Davare, Alberto Sangiovanni-Vincentelli, Sampada Sonalkar, Sri Kanajan, and Claudio Pinello. Design space exploration of automotive platforms in Metropolis. In *2005 SAE International*, Detroit, MI, USA, 2005.

[2] Xi Chen, Harry Hsieh, Felice Balarin, and Yosinori Watanabe. Automatic trace analysis for logic of constraints. In *DAC '03: Proceedings of the 40th conference on Design automation*, pages 460–465, New York, NY, USA, 2003. ACM Press.

[3] Tom M. Mitchell. Generalization as search. *Artificial Intelligence*, 18:203–226, 1982.

[4] Tessa A. Lau, Pedro Domingos, and Daniel S. Weld. Version space algebra and its application to programming by demonstration. In *17th International Conference on Machine Learning (ICML)*, pages 527–534, San Francisco, CA, 2000. Morgan Kaufmann.

[5] Tessa A. Lau, Pedro Domingos, and Daniel S. Weld. Learning programs from traces using version space algebra. In *International Conference on Knowledge Capture (K-CAP)*, pages 36–43, Banff, Alberta, Canada, 2003.