

% Extract nodes

% Constraints

### 2. ANSWER SET PROGRAMMING

Answer set programming (ASP) is a paradigm for declarative programming that effectively emerged in the late nineties.

- ► A rule-based language is used for problem encodings.
- Every program P, i.e., a set of rules, has a clearly defined semantics (the set of answer sets associated with P).
- The order of rules and the order of individual conditions in rules is irrelevant which gives a declarative nature for answer sets.
- Dedicated search engines—answer set solvers—are used to compute an answer set or several answer sets for a program.

© 2007 TKK / TCS

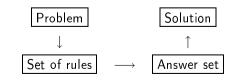
#### T-79.5102 / Autumn 2007

6

### Revising the Conceptual Model for ASP

Introduction

A problem is encoded so that the answer sets of the respective program and the solutions of the problem are in a tight correspondence.



Current answer set solvers expect *ground* programs as their input which implies a preprocessing step in order to remove variables.

## Example: Graph Coloring

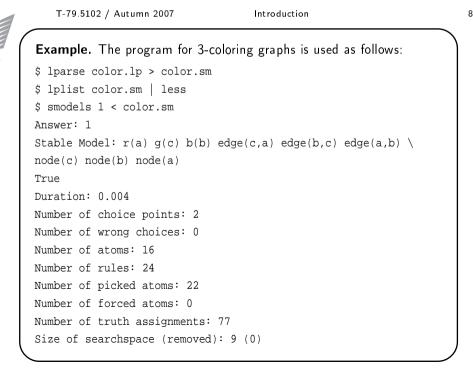
edge(a,b). edge(b,c). edge(c,a). % Edges

node(X) :- edge(X,Y). node(Y) :- edge(X,Y).

r(X) :- not g(X), not b(X), node(X). % Red g(X) :- not b(X), not r(X), node(X). % Green b(X) :- not r(X), not g(X), node(X). % Blue

:- r(X), r(Y), edge(X,Y).:- g(X), g(Y), edge(X,Y).:- b(X), b(Y), edge(X,Y).

 $\odot$  2007 TKK / TCS

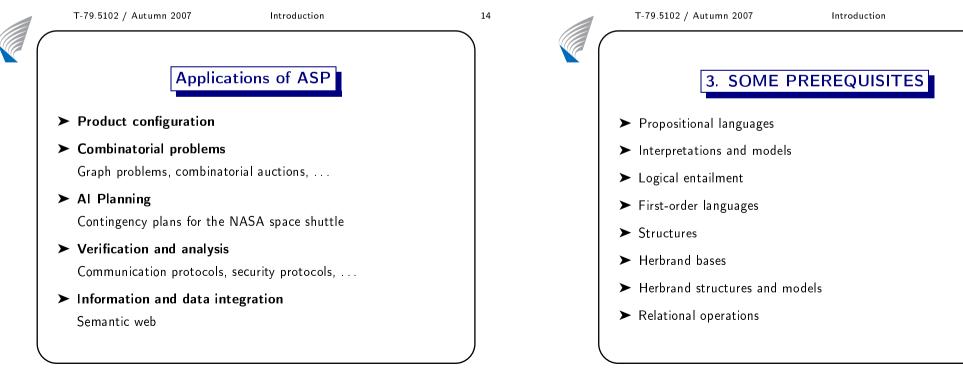


	T-79.5102 / Autumn 2007	Introduction			T-79.5102 / Autumn 2			Introduc	lion		
				<b>(</b> (	<b>Example.</b> Royle's Su	Doku pı	zzle wi	:h 16 c	lues ge	ts solved i	in 52 ms
	Roots of	of ASP		\$	lparse sudoku.lp	p royle	.lp	smodel	s 1		
		JIAJI		s	models version 2.	32. Rea	ading.	done			
	Knowledge representation and			A	nswer: 1						
	<ul> <li>Knowledge representation and</li> </ul>	reasoning		S	Stable Model: valu	ue(8,8,	l) val	1e(4,7	,1)	. value	(3,1,9)
	► Databases (SQL)			Г	rue						
	Deductive databases			E	Ouration: 0.052						
				N	Number of choice p	points:	1				
	<ul> <li>Logic programming (PROLOG)</li> </ul>			N	Number of wrong ch	noices:	0				
	<ul> <li>SLD-Resolution</li> </ul>			N	Number of atoms: 1	L156					
	<ul> <li>Negation as failure to prove</li> </ul>			N	Number of rules: 9	928					
	<ul> <li>Clark's completion and suppo</li> </ul>	orted models		N	Number of picked a	atoms:	321				
	<ul> <li>Constraint programming</li> </ul>			N	Number of forced a	atoms:	51				
				N	Number of truth as	ssignme	nts: 8	017			
				S	lize of searchspac	ce (rem	oved):	36 (0	)		
$\overline{\ }$											
	_					_					
	© 2007 T	KK / TCS				©	2007 TK	< / тсs			
	© 2007 T T-79.5102 / Autumn 2007	KK / TCS	10		T-79.5102 / Autumn 2	-	2007 TK	<pre>&lt; / TCS Introduce</pre>	tion		
			10		T-79.5102 / Autumn 2	-	2007 TK	·	tion		
$\left( \right)$	T-79.5102 / Autumn 2007		10			007		Introduc		acted from	n the
$\left( \right)$	T-79.5102 / Autumn 2007	Introduction	10		<b>Example.</b> The corres	007 ponding	solutio	Introduc	e extra		n the
	T-79.5102 / Autumn 2007 Example: Su	Introduction	10			007 ponding	solutio	Introduc	e extra		the
	T-79.5102 / Autumn 2007 Example: Su umber(1). number(2). number(3).	Introduction Doku Puzzle number(4). number(5).	10		Example. The corres	007 ponding visualized	solutio I as a s	Introduc n can b olved S	e extra uDoku	puzzle:	the
	T-79.5102 / Autumn 2007 Example: Su	Introduction Doku Puzzle number(4). number(5).	10		Example. The corres	007 ponding	solutio	Introduc n can b olved S	e extra	puzzle:	the
nı	T-79.5102 / Autumn 2007 Example: Su umber(1). number(2). number(3).	Introduction Doku Puzzle number(4). number(5).	10		Example. The corres	oo7 ponding visualized 9 3	solutio I as a s	Introduce n can b olved S	e extra uDoku	puzzle:	1 the
nı	T-79.5102 / Autumn 2007 <b>Example: Su</b> umber(1). number(2). number(3). umber(6). number(7). number(8).	Introduction Doku Puzzle number(4). number(5).	10		Example. The corres	oo7 ponding visualized 93 68	solutio l as a s 8 6 5 <b>3</b>	Introduce n can b olved S 7 4 2 9	e extra uDoku 2 <b>5</b> 1 7	puzzle:	the
nı bo	T-79.5102 / Autumn 2007 <b>Example: Su</b> umber(1). number(2). number(3). umber(6). number(7). number(8).	Introduction Doku Puzzle number(4). number(5). number(9).	10		Example. The corres inswer set and then v 1 4 7	007 ponding visualized 9 3 6 8 5 <b>2</b>	solutio as a s 8 6 5 <b>3</b> 1 <b>4</b>	Introduce n can b olved S 7 4 2 9 9 6	e extra uDoku 2 <b>5</b> 1 7 8 3	puzzle:	n the
nı bo re	T-79.5102 / Autumn 2007 <b>Example: Su</b> umber(1). number(2). number(3). umber(6). number(7). number(8). order(1). border(4). border(7). egion(X,Y) :- border(X), border	Introduction Doku Puzzle number(4). number(5). number(9). (Y).	10		Example. The corres inswer set and then v 1 4 7 6	007 ponding visualized 9 3 6 8 5 <b>2</b> 6 2 1	solutio as a s 8 6 5 <b>3</b> 1 <b>4</b> 4 7	Introduce n can b olved S 7 4 2 9 9 6 3 5	e extra uDoku 2 <b>5</b> 1 7 8 3 9 8	puzzle:	n the
nı bo re	T-79.5102 / Autumn 2007 <b>Example: Su</b> umber(1). number(2). number(3). umber(6). number(7). number(8). order(1). border(4). border(7). egion(X,Y) :- border(X), border { value(X,Y,N):number(X):number	Introduction Doku Puzzle number(4). number(5). number(9). (Y). r(Y):	10		Example. The corres inswer set and then v 1 4 7 6	007 ponding visualized 9 3 6 8 5 <b>2</b>	solutio as a s 8 6 5 <b>3</b> 1 <b>4</b>	Introduce n can b olved S 7 4 2 9 9 6 3 5	e extra uDoku 2 <b>5</b> 1 7 8 3	puzzle:	n the
nı bo re	T-79.5102 / Autumn 2007 <b>Example: Su</b> umber(1). number(2). number(3). umber(6). number(7). number(8). order(1). border(4). border(7). egion(X,Y) :- border(X), border { value(X,Y,N):number(X):number X1<=X:X<=X1+2:Y1<=Y:Y<=Y1+2	Introduction Doku Puzzle number(4). number(5). number(9). (Y). r(Y):	10		Example. The corresonswer set and then v 1 4 7 6 5	007 ponding visualized 9 3 6 8 5 <b>2</b> 6 2 1	solutio as a s 8 6 5 <b>3</b> 1 <b>4</b> 4 7	Introduce n can b olved S 7 4 2 9 9 6 3 5 8 <b>7</b>	e extra uDoku 2 <b>5</b> 1 7 8 3 9 8	puzzle:	n the
nı bo re	T-79.5102 / Autumn 2007 <b>Example: Su</b> umber(1). number(2). number(3). umber(6). number(7). number(8). order(1). border(4). border(7). egion(X,Y) :- border(X), border { value(X,Y,N):number(X):number	Introduction Doku Puzzle number(4). number(5). number(9). (Y). r(Y):	10		Example. The correst inswer set and then v 1 4 7 6 5 9	007 ponding visualized 9 3 6 8 5 2 5 2 1 3 4	solutio as a s 8 6 5 <b>3</b> 1 <b>4</b> 4 7 9 1	Introduce n can b olved S 7 4 2 9 6 3 5 8 7 6 3	e extra uDoku 2 <b>5</b> 1 7 8 3 9 8 6 2	puzzle:	the
nu bo re 1	<pre>T-79.5102 / Autumn 2007</pre>	<pre>Introduction Doku Puzzle number(4). number(5). number(9). (Y). r(Y): } 1</pre>	10		Example. The corres inswer set and then v 1 4 7 6 5 9 2	007 ponding visualized 9 3 6 8 5 2 6 2 1 6 2 1 6 3 4 8 7 2 1 6	solutio as a s 8 6 5 <b>3</b> 1 <b>4</b> 4 7 9 1 <b>2</b> 5 3 9	Introduct           n can b           olved S           7         4           2         9           9         6           3         5           6         3           5         8	e extra uDoku 2 <b>5</b> 1 7 8 3 9 8 6 2 4 <b>1</b> 7 4	puzzle:	the
nı bo 1	<pre>T-79.5102 / Autumn 2007</pre>	<pre>Introduction Doku Puzzle number(4). number(5). number(9). (Y). r(Y): } 1 umber(X), number(Y).</pre>	10		Example. The corresonswer set and then value of the set and the value of the set of the	007 ponding visualized 9 3 6 8 5 2 1 3 4 9 8 7 2 1 6 5 7 5	solutio as a s 3 4 4 7 9 1 <b>2</b> 5 3 9 6 2	Introduct         n can b         olved S         7       4         2       9         9       6         3       5         8       7         6       3         5       8         4       1	e extra uDoku 2 <b>5</b> 1 7 8 3 9 8 6 2 4 <b>1</b> 7 4 <b>3</b> 9	puzzle:	n the
nı bo 1 :-	<pre>T-79.5102 / Autumn 2007</pre>	<pre>Introduction Doku Puzzle number(4). number(5). number(9). (Y). r(Y): } 1 umber(X), number(Y). umber(N), number(X).</pre>			Example. The corresonswer set and then value of the set and the value of the set of the	007 ponding visualized 9 3 6 8 5 2 6 2 1 6 2 1 6 3 4 8 7 2 1 6	solutio as a s 8 6 5 <b>3</b> 1 <b>4</b> 4 7 9 1 <b>2</b> 5 3 9	Introduct         n can b         olved S         7       4         2       9         9       6         3       5         8       7         6       3         5       8         4       1	e extra uDoku 2 <b>5</b> 1 7 8 3 9 8 6 2 4 <b>1</b> 7 4	puzzle:	the

**Example.** Actually, there are 2 solutions for this 16 clue puzzle. The other is obtained by exchanging the occurrences of 8 and 9:

1	8	3	9	6	7	4	2	5
4	6	9			2			7
7	5	2	1	4	8	6	9	3
6	2	1	4	7	3	5	8	9
5	3	4	8	1	9	7	6	2
8	9	7	2	5	6	3	4	1
2	1	6	3	8	5	9	7	4
9	7	5	6	2	4	1	3	8
3	4	8	7	9	1	2	5	6

© 2007 TKK / TCS



13

# Factors Behind the Success of ASP

- > The performance of computers has increased remarkably.
- ► Implementation techniques have advanced rapidly.
- Many efficient solvers are publicly available: smodels, clasp, cmodels, dlv, ...
- Rule-based languages are highly expressive—enabling concise encodings for a wide variety of problems.
- ASP languages lend themselves to fast prototyping with little programming effort.

© 2007 TKK / TCS

16

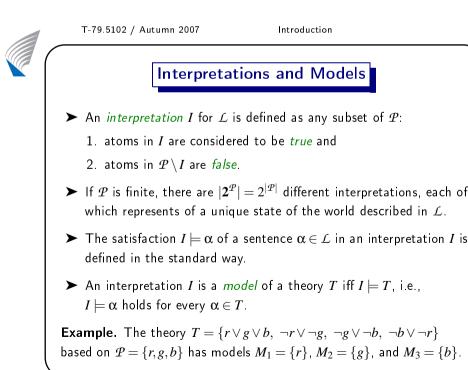
18

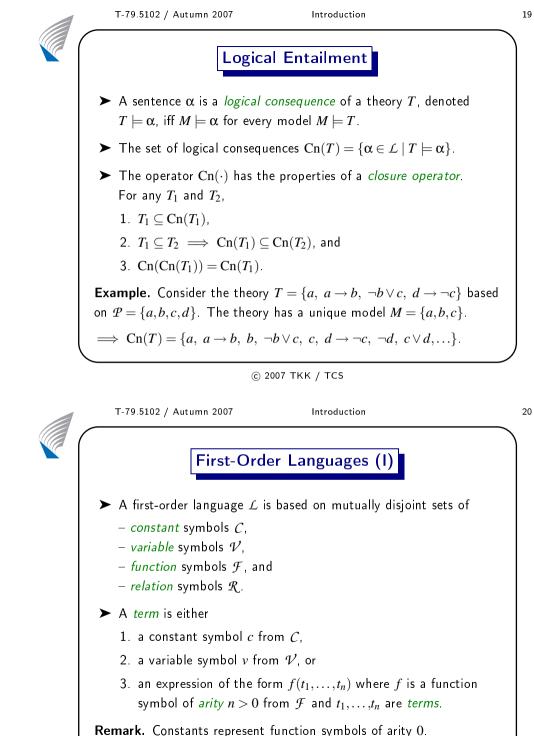
## **Propositional Languages**

- > Any set of *atomic sentences*  $\mathcal{P} \neq \emptyset$ , or *atoms* for short, induces a propositional language  $\mathcal{L}$ — the set of well-formed sentences.
- $\blacktriangleright$  Sentences are built using the atoms of  $\mathcal{P}$  and propositional connectives  $\neg$  (negation),  $\land$  (conjunction),  $\lor$  (disjunction),  $\rightarrow$ (implication), and  $\leftrightarrow$  (equivalence).
  - 1 Atomic sentences are *sentences*.
  - 2. If  $\alpha$  and  $\beta$  are sentences, then expressions of the forms  $(\neg \alpha)$ ,  $(\alpha \lor \beta), (\alpha \land \beta), (\alpha \to \beta), (\alpha \leftrightarrow \beta)$  are also sentences.
- $\blacktriangleright$  Propositional theories T are defined as subsets of  $\mathcal{L}$ .

**Example.** The theory  $T = \{r \lor g \lor b, \neg r \lor \neg g, \neg g \lor \neg b, \neg b \lor \neg r\}$ describes the 3-coloring of a single node in a graph.

© 2007 TKK / TCS





T-79.5102 / Autumn 2007

24



- > Atomic formulas R,  $t_1 = t_2$ , and  $R(t_1, ..., t_n)$  are satisfied by S iff  $\langle \rangle \in \mathbb{R}^S$ ,  $(t_1)^S = (t_2)^S$ , and  $\langle (t_1)^S, \dots, (t_n)^S \rangle \in \mathbb{R}^S$ , respectively.
- $\blacktriangleright$  The satisfaction of a first order formula/sentence  $\alpha$  in a structure is defined in the standard way.
- > Structures that are *models* of sentences  $(S \models \alpha)$  and theories  $(S \models T)$  are distinguished in analogy to propositional logic.
- > Moreover, the definition of  $T \models \alpha$ , i.e., whether a sentence  $\alpha$  is a logical consequence of a theory T, remains unchanged.

**Example.** For  $T = \{E(0), \forall x(E(x) \rightarrow O(s(x))), \forall x(O(x) \rightarrow E(s(x)))\}$ formalizing even natural numbers:  $T \models E(s(s(0)))$  but  $T \not\models \neg E(s(0))$ .

© 2007 TKK / TCS

T-79.5102 / Autumn 2007 Introduction **Herbrand Bases** > A ground term is a term having no occurrences of variables.  $\blacktriangleright$  Given the sets C and  $\mathcal{F}$  (see above), the Herbrand universe is the set of ground terms constructible using the symbols of C and  $\mathcal{F}$ .  $\blacktriangleright$  Given the set  $\mathcal{R}$ , the Herbrand base consists of atomic sentences  $R(t_1,\ldots,t_n)$  where  $R \in \mathcal{R}$  is of arity *n* and each  $t_i$  is a ground term. > A Herbrand instance of an atomic formula  $R(t_1,...,t_n)$  is obtained by substituting ground terms for variables occurring in  $t_1, \ldots, t_n$ .  $\blacktriangleright$  We may also define the Herbrand base Hb(T) of a theory T by inspecting which constant/function symbols occur in T. **Example.** For the previous theory T formalizing even natural numbers, we have  $Hb(T) = \{E(0), O(0), E(s(0)), O(s(0)), \ldots\}$ .

First-Order Languages (II) ► An *atomic formula* is an expression of the form 1. R for each relation symbol of arity 0 from  $\mathcal{R}$ . 2.  $t_1 = t_2$  where  $t_1$  and  $t_2$  are terms, or 3.  $R(t_1, \ldots, t_n)$  where R is a relation symbol of arity n > 0 from  $\mathcal{R}$ . and  $t_1, \ldots, t_n$  are terms. ► Atomic formulas are *formulas*.  $\blacktriangleright$  If  $\alpha$  and  $\beta$  are formulas and v is a variable from  $\mathcal{V}$ , then expressions of the forms  $(\neg \alpha), (\alpha \lor \beta), (\alpha \land \beta), (\alpha \to \beta), (\alpha \leftrightarrow \beta), (\forall v\alpha), and (\exists v\alpha)$ are also formulas ► A sentence is a formula having no free occurrences of variables. © 2007 TKK / TCS T-79.5102 / Autumn 2007 22 Introduction Structures (I)  $\blacktriangleright$  An interpretation for a first-order language  $\mathcal{L}$  is a structure S based on a *universe* U which is any non-empty set and 1. each  $c \in C$  is mapped to an element  $c^{S} \in U$ . 2. each  $v \in \mathcal{V}$  is mapped to an element  $v^S \in U$ , 3. each  $f \in \mathcal{F}$  is mapped to a function  $f^S: U^n \to U$  where *n* is the arity of f, and 4. each  $R \in \mathcal{R}$  with an arity *n* is mapped to a relation  $R^S \subseteq U^n$ .  $\blacktriangleright$  Given a structure S, each term t is mapped to an element  $t^s \in U$ . **Example.** Given a constant symbol 0 and a *unary* (of arity 1) function symbol s we may define a structure S based on  $U = \{0, 1, 2, ...\}$  by setting  $0^S = 0$  and  $s^S : x \mapsto x+1$ . Thus  $(s(s(s(0))))^S = 3$ .

 $\blacktriangleright$  A Herbrand structure S is based on a fixed interpretation of

1. Each  $c \in C$  is mapped to  $c^S = c$ .

satisfies all sentences of T.

constant and function symbols over the Herbrand universe:

2. Each  $f \in \mathcal{F}$  is mapped to  $f^S : \langle t_1, \ldots, t_n \rangle \mapsto f(t_1, \ldots, t_n)$ .

 $\blacktriangleright$  Any Herbrand structure S can be viewed as a propositional

interpretation  $I = \{R(t_1, \dots, t_n) \in \operatorname{Hb}(T) \mid S \models R(t_1, \dots, t_n)\}$ .

 $\blacktriangleright$  A Herbrand model M of a theory T is a Herbrand structure that

**Example.** For the theory T formalizing even natural numbers, we have a Herbrand model  $M = \{E(0), O(s(0)), E(s(s(0))), O(s(s(s(0)))), \ldots\}$ .

© 2007 TKK / TCS

 $\implies$  Only interpretations of variables and predicates can vary!

27

28

## OBJECTIVES

- ➤ You have the necessary premises for the course, i.e., you are familiar with the syntax and semantics of classical logic.
- You know the main characteristics of declarative programming languages and understand the difference w.r.t. procedural ones.
- > You understand the conceptual model of answer set programming.
- You are able to list the basic steps which are required to to apply ASP in declarative problem solving.

 $\odot$  2007 TKK / TCS

T-79.5102 / Autumn 2007 26 T-79.5102 / Autumn 2007 Introduction Introduction **Relational operations** TIME TO PONDER Assume that  $R_1$  and  $R_2$  are binary relations (of arity 2) over a fixed **Definition.** The set of classical models associated with a propositional universe U, i.e.,  $R_1 \subseteq U^2$  and  $R_2 \subseteq U^2$ . theory T is  $CM(T) = \{M \subseteq Hb(T) \mid M \models T\}.$ 1. The *union* of  $R_1$  and  $R_2$  is **Problem.** Let  $T_1$  and  $T_2$  be arbitrary propositional theories.  $R_1 \cup R_2 = \{ \langle x, y \rangle \in U^2 \mid \langle x, y \rangle \in R_1 \text{ or } \langle x, y \rangle \in R_2 \}.$ Which one of the following is correct in general? 2. The *intersection* of  $R_1$  and  $R_2$  is 1.  $\operatorname{CM}(T_1 \cup T_2) = \operatorname{CM}(T_1) \cap \operatorname{CM}(T_2)$ .  $R_1 \cap R_2 = \{ \langle x, y \rangle \in U^2 \mid \langle x, y \rangle \in R_1 \text{ and } \langle x, y \rangle \in R_2 \}.$ 2.  $CM(T_1 \cup T_2) = \{M_1 \cup M_2 \mid M_1 \in CM(T_1) \text{ and } M_2 \in CM(T_2)\}.$ 3. The projections of  $R_1$  w.r.t. the first/second arguments are 3.  $CM(T_1 \cup T_2) =$  $P_1 = \{x \in U \mid \langle x, y \rangle \in R_1\}$  and  $P_2 = \{y \in U \mid \langle x, y \rangle \in R_1\}.$  $\{M_1 \cup M_2 \mid M_1 \in CM(T_1), M_2 \in CM(T_2), \text{ and } M_1 \cap H = M_2 \cap H\}$ 4. The *composition* of  $R_1$  of  $R_2$  is where  $H = \text{Hb}(T_1) \cap \text{Hb}(T_2)$ .  $R_1 \circ R_2 = \{ \langle x, y \rangle \in U^2 \mid \langle x, z \rangle \in R_1 \text{ and } \langle z, y \rangle \in R_2 \}.$ 

### © 2007 TKK / TCS

#### © 2007 TKK / TCS