The Hunter-Gatherer method

(Beale 1997)

Effective measures to reduce combinatorics

Application to issues in knowledge-intensive machine translation

AI TECHNIQUES USED

Motivation

- **Millions of combinations theoretically possible, but**
- **Dependencies limited**

Techniques

- **Branch and bound – local optimization; hunt down non-optimal, impossible**
- **Constraint systems – circuits of interdependencies**
- **Solution synthesis – gather together optimal partial solutions**

Application

- **Computational semantic processing**
- **Text planning converted into a constraint-satisfaction problem**
- **Large scale spanish-english MT system (New Mexico State Univ.)**

COMPLEXITY OF SEMANTICS

Exportation de Brasil a paises de Union Europea ascender a 2,395 milliones de dolares

Especially prepositions can have many senses:

“de”: OWNED_BY, LOC, TEMP, MADE_OF, INSTR, RELATION, SOURCE

“a”: LOC, THEME, MEASUREMENT, RELATION, DESTINATION

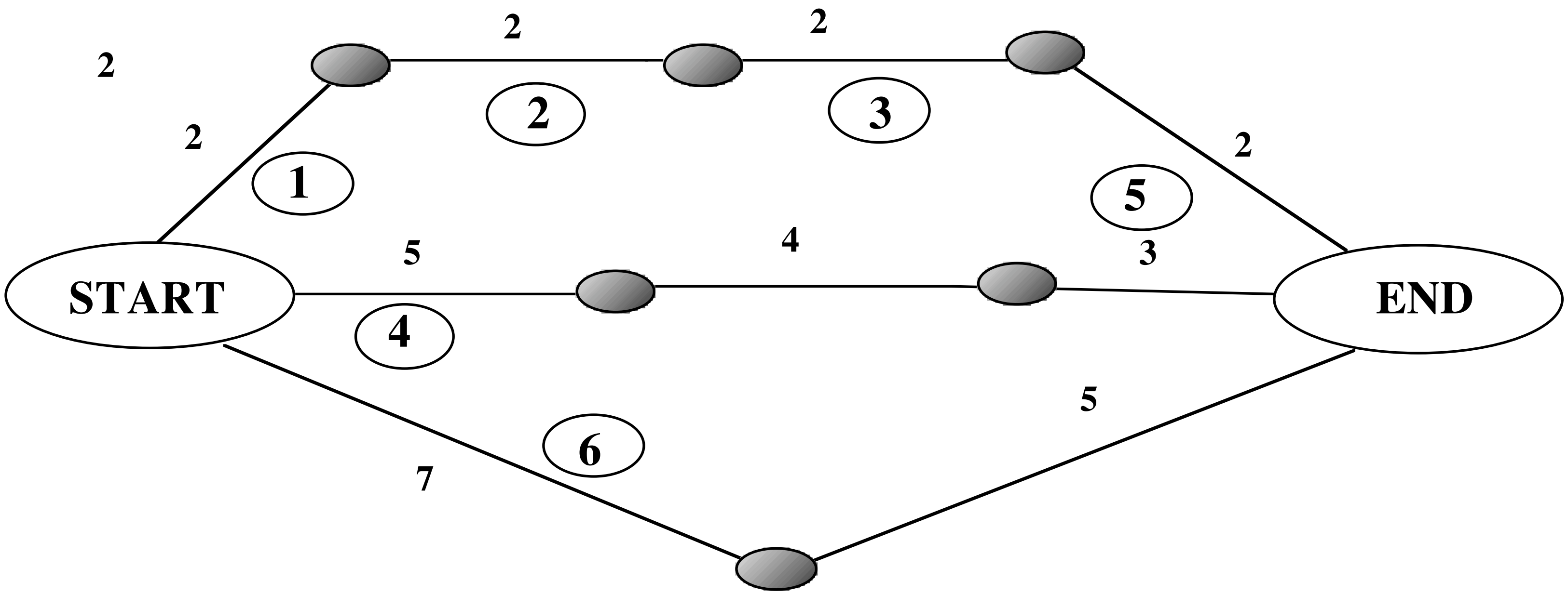
Complexity

- **Noun phrases (approx. 500 for “Exportation … Europea”)**
- **Clauses (approx. 18,000 for “Exportation … dolares”)**
- **Sentences (average over 50 millions according to NMSU corpus)**

Idea

- **Partitioning the problem into relatively independent subproblems**
- **Attacking them separately and combining solution candidates**
- **Result – guarantee of optimal answer in near-linear time**

BRANCH AND BOUND



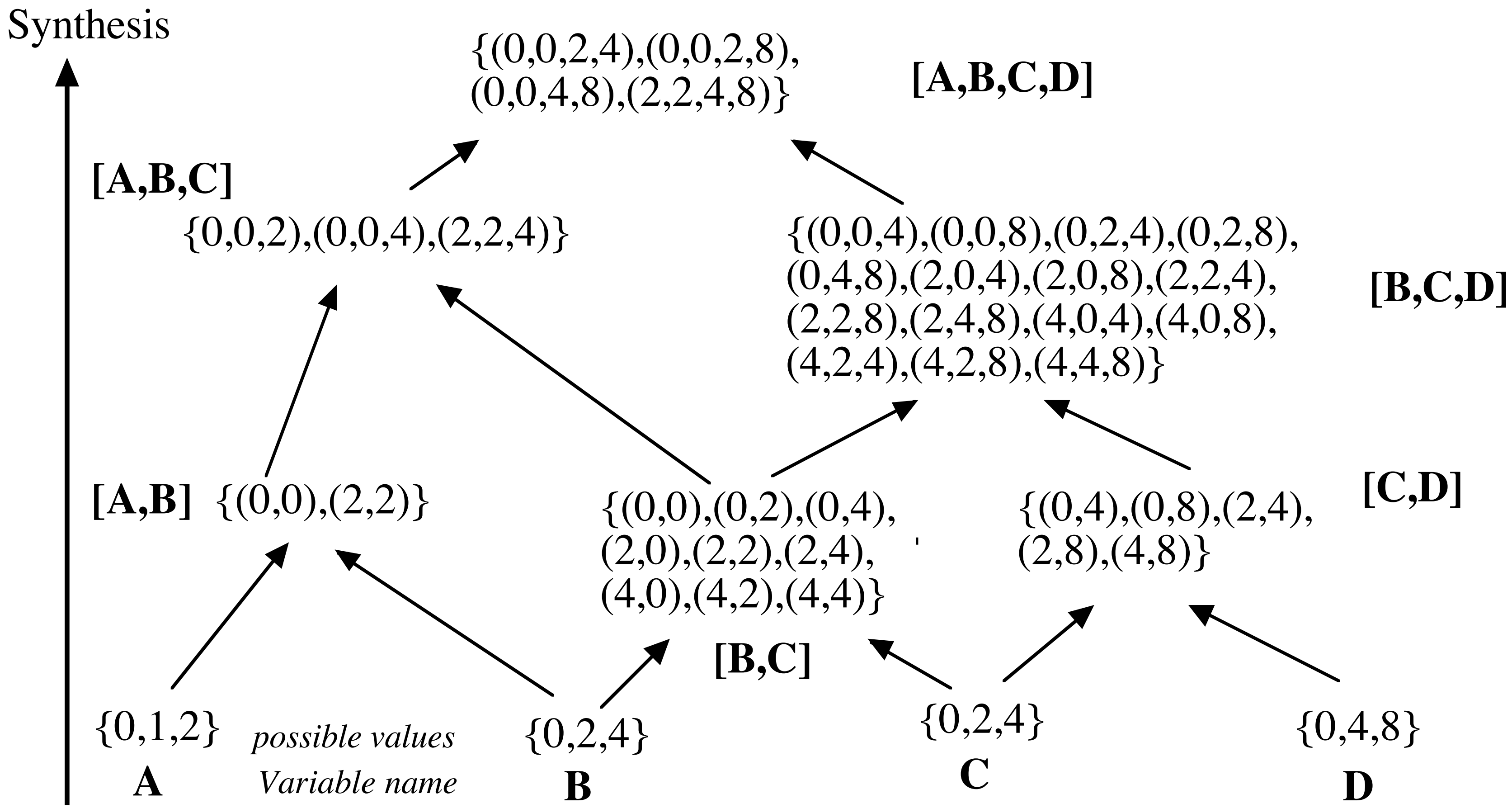
CONSTRAINT SATISFACTION

A = {0,1,2} B = {1,2,3} C = {1,2} A = B A < C

A = 0
- **B = 1**
  - **C = 1 {0,1,1} A ≠ B**
  - **C = 2 {0,1,2} A ≠ B** — **Backtracking required**
- **B = 2**
  - **C = 1 {0,2,1} A ≠ B**
  - **C = 2 {0,2,2} A ≠ B** — **Inconsistent partial combinations**
- **B = 3**
  - **C = 1 {0,3,1} A ≠ B**
  - **C = 2 {0,3,2} A ≠ B** — **Inconsistent partial combinations repeated**

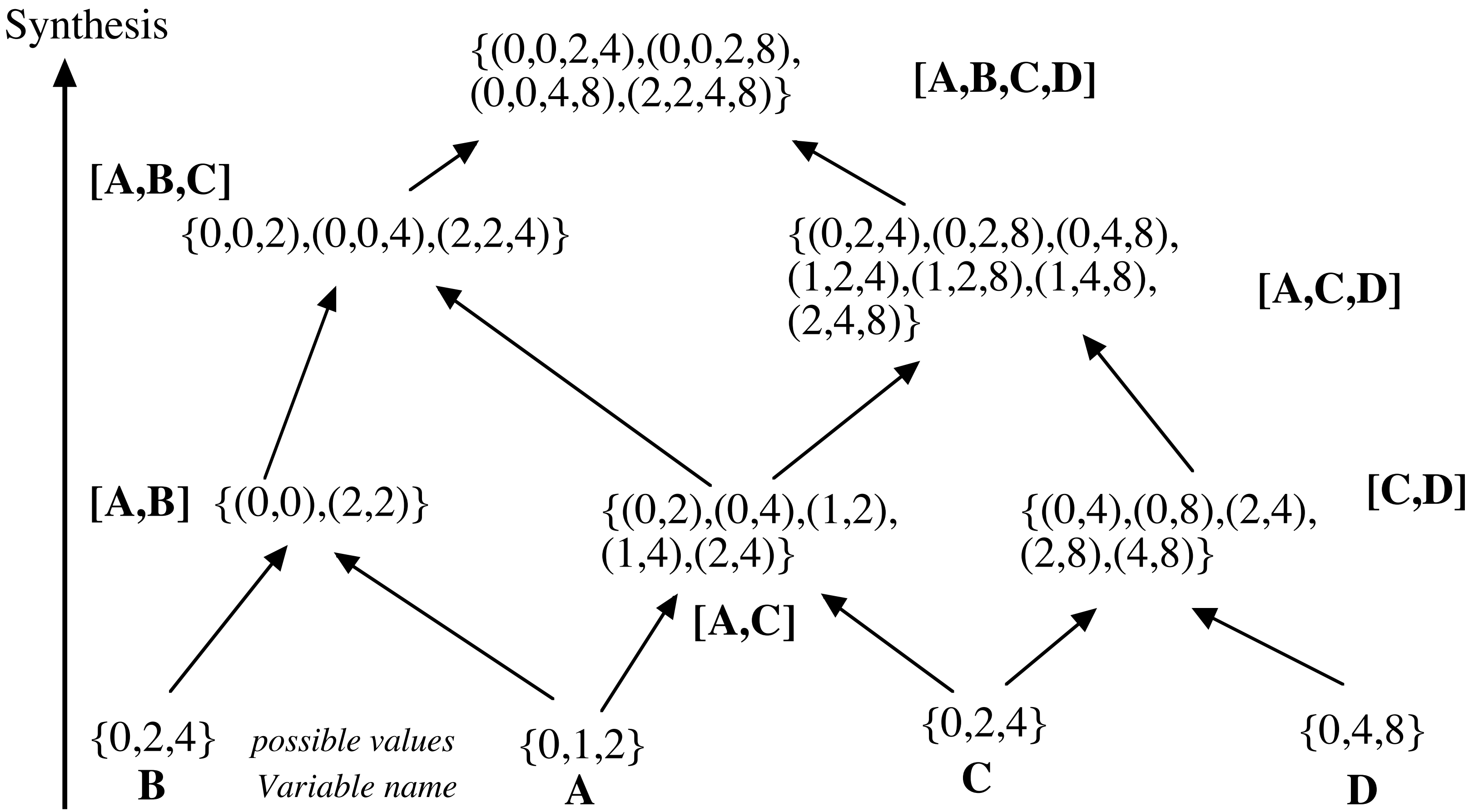
A = 1
- **B = 1**
  - **C = 1 {1,1,1} A ≥ C**
  - **C = 2 {1,1,2} OK**

SOLUTION SYNTHESIS (1)



Constraints: {A = B}, {C < D} {A < C}

SOLUTION SYNTHESIS (2) – BETTER ORDERING



Constraints: {A = B}, {C < D} {A < C}

CONSISTENCY STATES

Node consistency

Domains of each variable reduced to set of possible values

Satisfying unary constraints

Arc consistency

Domains of each variable reduced to set of possible values

Satisfying binary constraints connecting any two nodes

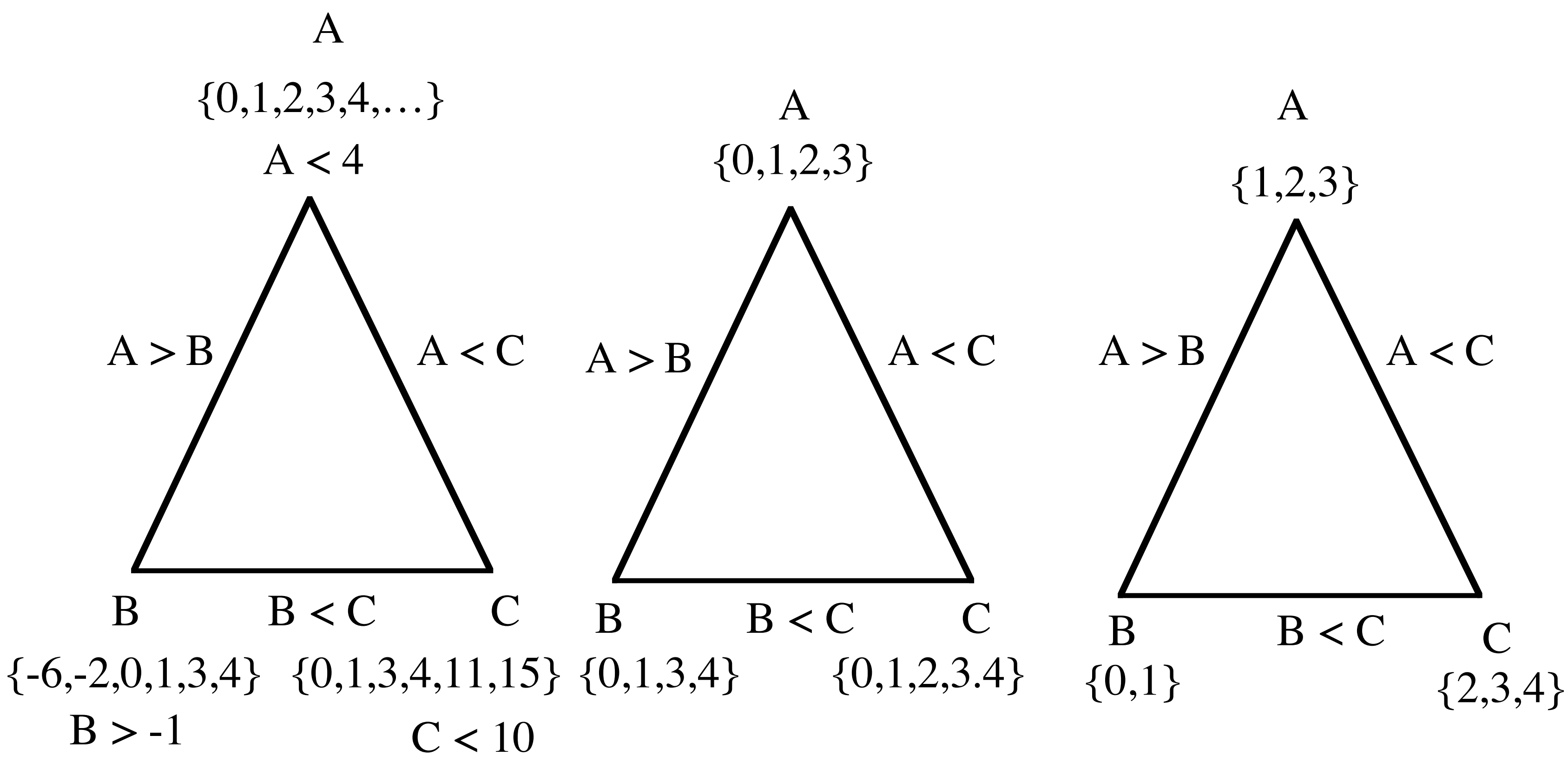
Path consistency

Eliminates impossible partial solutions

Computationally very expensive

Alternative: dynamic application of arc consistency

CONSISTENCY STATES – EXAMPLES



A.Unconstrained Graph B.Node-consistent Graph C.Arc-consistent Graph

METHODS USED FOR CONSTRAINT SATISFACTION

Linear programming (simplex)

Very powerful, but problems with initialization, looping, termination

Non-linear, non-decomposable, non-dynamic

Non-serial dynamic programming

Eliminates one variable at a time (Gaussian-like substitutions)

Builds a chain of intermediate functions, stored in a look-up table

Hunter gatherer

Decomposes a problem into subgraphs

thereby builds blocks of variables

Can better deal with changes/additions to the problem definition (locally)

APPLICATION TO COMPUTATIONAL SEMANTICS

Representation

Word sense interpretations as unary constraints

Relations among adjacent words as binary constraints

Using plausibility measures for interpretations (metonymy, metaphor)

Techniques

Decomposing a problem into subgraphs according to constraint information

Ordering to guide solution synthesis by using circles

Branch-and-bound to filter non-optimal solutions

to prevent combinatorial explosion

SOLUTION SYNTHESIS ALGORITHM (Tsang and Foster, 1990)

Basic technique

Make local assignments that are consistent, building partial solutions

Combine simple partial solutions into more complex ones incrementally

Improvements

Propagate constraints to eliminate inconsistent partial solutions

Combine only “adjacent” partial solutions into more complex ones (ordering!)

Advantages and disadvantages

+ Limiting the number of solution sets, potential for parallel implementations

- Limiting propagation chances (e.g., constraints between “distant” variables)

SOLUTION SYNTHESIS – AN EXAMPLE (1)

Translating: ***"IBM acquired Jacob-Smith for ten-million-dollars".***

WORD	CONCEPT	CONSTRAINTS	EXAMPLE
IBM(I)			
	ORG		
acquired(A)			
	TAKE-OVER(T-O)	[I=ORG J=ORG]	
	OBTAIN(OBT)	[I=ANIMATE J=INANIMATE]	
Jacob-Smith(J)			
	HUMAN(HUM)		
	ORG		
for(F)			
	COST	[A=EVENT T=MONEY]	I bought it for 10 dollars.
	BENEFIC(BEN)	[A=EVENT T=ANIMAL]	I bought it for Sam.
	PURPOSE(PUR)	[A=EVENT T=EVENT]	I bought it for mowing the lawn.
	DURATION(DUR)	[A=EVENT T=TIME]	I hid it for 10 hours.
ten-million-dollars(T)			
	MONEY(MON)		

SOLUTION SYNTHESIS – AN EXAMPLE (2)

IAJFT — **5th order solution sets**

IAJF AJFT — **4th order solution sets**

IAJ AJF JFT — **3rd order solution sets**

IA AJ JF FT — **2nd order solution sets**

I A J F T — **1st order solution sets**

"IBM" "acquired" "Jacob-Smith" "for" "ten-million-dollars"

Order 1 nodes:

N_I = {(<I,ORG>)}

N_A = {(<A,T-O>),(<A,OBT>)}

N_J = {(<J,HUM>),(<J,ORG>)}

N_F = {(<F,COST>),(<F,BEN>),(<F,PUR>),(<F,DUR>)}

N_T = {(<T,MON>)}

SOLUTION SYNTHESIS – AN EXAMPLE (3)

		IAJFT			5th order solution sets
	IAJF		AJFT		4th order solution sets
	IAJ	AJF	JFT		3rd order solution sets
IA	AJ		JF	FT	2nd order solution sets
I	A	J	F	T	1st order solution sets
"IBM"	*"acquired"*	*"Jacob-Smith"*	*"for"*	*"ten-million-dollars"*	

Order 2 nodes:

N_{IA} = {(ORG,T-O),(ORG,OBT)}

N_{AJ} = {(T-O,ORG),(OBT,ORG)};; eliminates (T-O,HUM),(OBT,HUM)

**N_{JF} = {(HUM,COST),(HUM,BEN),(HUM,PUR),(HUM,DUR),
(ORG,COST),(ORG,BEN),(ORG,PUR),(ORG,DUR)}**

N_{FT} = {(COST,MON)};; eliminates (BEN,MON),(PUR,MON),(DUR,MON)

SOLUTION SYNTHESIS – AN EXAMPLE (4)

			IAJFT			**5th order solution sets**
			IAJF AJFT			**4th order solution sets**
		IAJ	AJF	JFT		**3rd order solution sets**
	IA	AJ	JF	FT		**2nd order solution sets**
I	A		J	F	T	**1st order solution sets**

"IBM" "acquired" "Jacob-Smith" "for" "ten-million-dollars"

Order 3 nodes:

N_{IAJ} = {(ORG,T-O,ORG),(ORG,OBT,ORG)}

N_{AJF} = {(T-O,ORG,COST),(T-O,ORG,BEN),(T-O,ORG,PUR),(T-O,ORG,DUR),
(OBT,ORG,COST),(OBT,ORG,BEN),(OBT,ORG,PUR),(OBT,ORG,DUR)}

N_{JFT} = {(HUM,COST,MON),(ORG,COST,MON)}

SOLUTION SYNTHESIS – AN EXAMPLE (5)

		IAJFT			**5th order solution sets**
		IAJF	AJFT		**4th order solution sets**
	IAJ	AJF	JFT		**3rd order solution sets**
IA	AJ	JF	FT		**2nd order solution sets**
I	A	J	F	T	**1st order solution sets**

"IBM" "acquired" "Jacob-Smith" "for" "ten-million-dollars"

Order 4 nodes:

N_{IAJF} = {(ORG,T-O,ORG,COST),(ORG,T-O,ORG,BEN),(ORG,T-O,ORG,PUR), (ORG,T-O,ORG,DUR), (ORG,OBT,ORG,COST),(ORG,OBT,ORG,BEN), (ORG,OBT,ORG,PUR),(ORG,OBT,ORG,DUR)}

N_{AJFT} = {(T-O,ORG,COST,MON),(OBT,ORG,COST,MON)}

SOLUTION SYNTHESIS – AN EXAMPLE (6)

		IAJFT			**5th order solution sets**
		IAJF	**AJFT**		**4th order solution sets**
	IAJ	**AJF**	**JFT**		**3rd order solution sets**
IA	**AJ**	**JF**	**FT**		**2nd order solution sets**
I	**A**	**J**	**F**	**T**	**1st order solution sets**
"IBM"	***"acquired"***	***"Jacob-Smith"***	***"for"***	***"ten-million-dollars"***	

Solution set:

N_{IAJFT} = {(ORG,T-O,ORG,COST,MON),(ORG,OBT,ORG,COST,MON)}

SOLUTION SYNTHESIS – AN EXAMPLE (7)

Incorportaing propagation

Assignments incompatible with all assignments to “adjacent” variable excluded

propagated to distant assignments

Examples

No reading of “acquire” fits to the reading of “Jacob-Smith” as human

All readings of “for” except to cost incompatible with “10 million dollar”, yields

N_{IA} = {(ORG,T-O),(ORG,OBT)}

N_{AJ} = {(T-O,ORG),(OBT,ORG)}

N_{JF} = {(ORG,COST),

N_{FT} = {(COST,MON)}

N_{IAJ} = {(ORG,T-O,ORG),(ORG,OBT,ORG)}

N_{AJF} = {(T-O,ORG,COST), (OBT,ORG,COST)}

N_{JFT} = {(ORG,COST,MON)}

N_{IAJF} = {(ORG,T-O,ORG,COST),(ORG,OBT,ORG,COST)}

N_{AJFT} = {(T-O,ORG,COST,MON),(OBT,ORG,COST,MON)}

MIKROKOSMOS MACHINE TRANSLATION SYSTEM

Complexity handling

Constraining complexity by taking into account dependencies

Linguistic problems are typically composed of subproblems

Microtheories

Meaning of natural language texts in a language-neutral interlingua

Input text represented as an element of a model of the world (ontology)

Lexicon represents meanings of open-class words as mappings
into ontological concepts

Separate microtheories handle non-propositional components of text meaning
speech acts, speaker attitude, relations among text units, deictic references

REPRESENTATION COMPONENTS

Text menaing representation (TMR)

Lexico-semantic dependencies

Stylistic factors, discourse relations, …

Instantiating, combining, and constraining concepts from the ontology

Ontology

Supplies world knowledge to lexical, syntactic, and semantic processes

Concepts typically have 5 to 10 slots linking them to other concepts

Application: company mergers and acquisition

- **> 5000 concepts**
- **Depth 10 or more along some paths**
- **Top level distinctions very stable (object, event, property)**

SEMANTIC LEXICON

SYN-STRUC zone

Specifications for syntactic parsing: subcategorization, complements allowed, …

Syntactic relationships are link to maning patterns (compositionality)

Variable bindings according to structural dependencies of lexeme

Syntactic pattern required may result in exclusion of aword sense

SEM zone

Underspecified TMR fragment with information according to a word extracted

Language-specific semantic constraints

- **May override those from the ontology or add to them**
- **Variety in lexemes is richer than the concepts in the ontology**

SEMANTIC ANALYSIS

Task

Combines knowledge contained in the ontology and lexicon in view of input

Retrieve semantic constraints, test each in context, and construct output

Generating constraints

List of constraints (possible sources):

1. ontological definition of word sense restricts semantics of its slot fillers

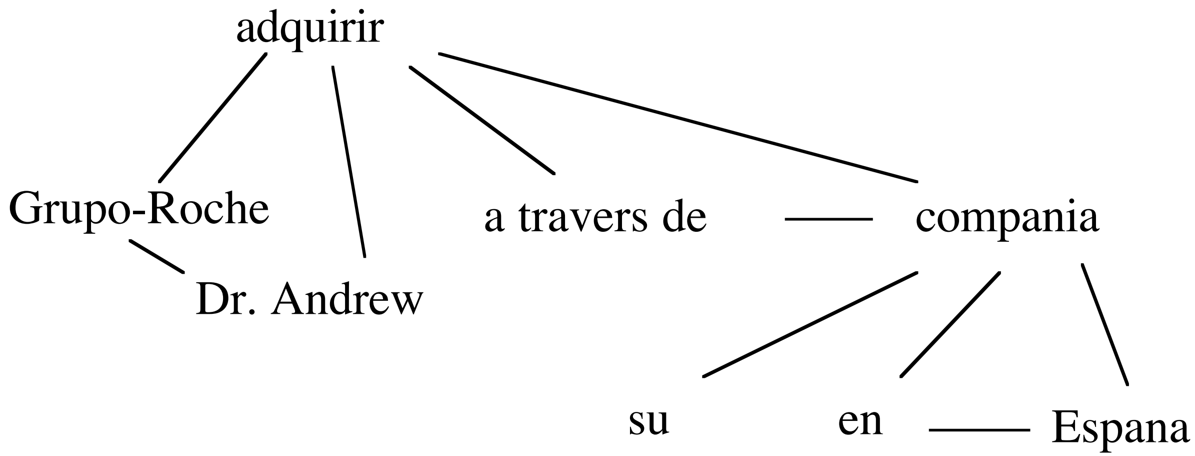
2. ontological definition restricts the slot it may be the filler of

3. ontological definition of slots (domain and range); may be very general

4. lexicon entry may include constraints that override or add to the ontology

5. other structures in the sentence that modify some word; e.g., adjectives

DETERMINING THE BEST COMBINATION OF SENSES



1. **“a travers de” is INSTRUMENT (LOCATION requires filling a PHYSICAL-OBJECT)**
2. **“en” is LOCATION (TEMPORAL requires its filler to be TEMPORAL-OBJECT)**
3. **“adquirir” maps onto ACQUIRE (LEARNING requires INFORMATION as THEME)**
4. **“Dr. Andrew” is an ORGANIZATION (HUMAN cannot be THEME of ACQUIRE)**
5. **“compania” not yet resolved between CORPORATION and SOCIAL-EVENT (would require restrictions on the INSTRUMENT slot of ACQUIRE)**

IDENTIFYING SUBGRAPHS

Building seeds

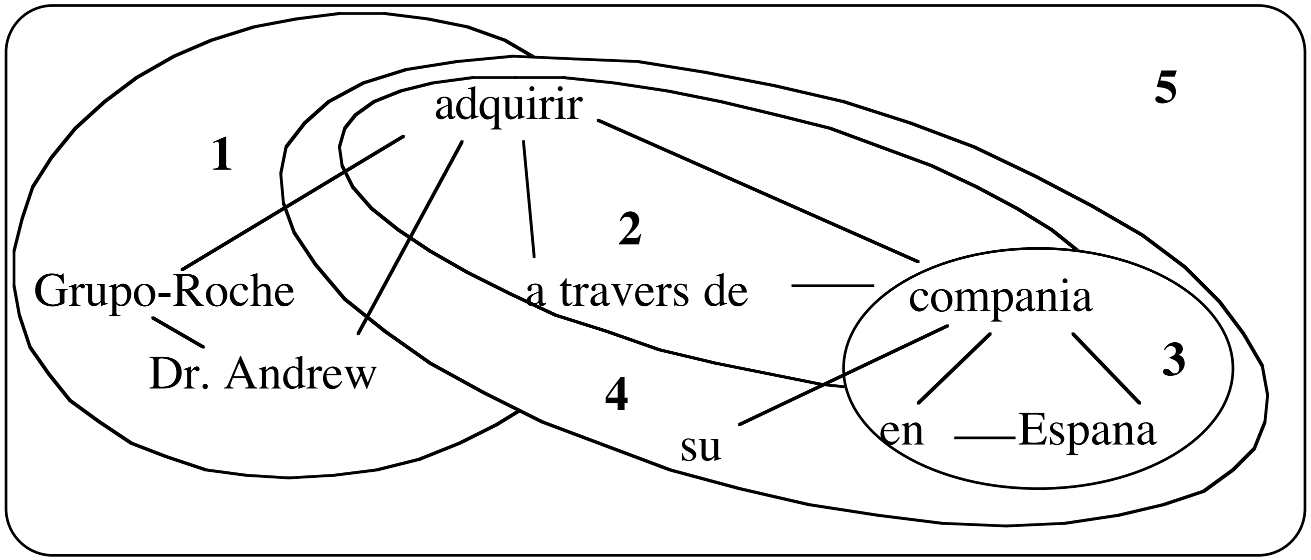
1. For each variable – set of variables adjacent to it
   (the set of variables constraining it directly)
2. Ordering seeds according to size (eliminating duplicates)
3. Build regions of a seed are the seeds plus variables adjacent to them
4. Take first, and subsequent ones if independent of all previous ones
5. Action to expand the seed
   (no additional constraints on a variable in the seed
   or combining constraints of a variable from two seeds)
6. Proceed with step 5 until all variables are covered

Better partitionings possible by following the semantic tree structure

CREATING CIRCLES (SUBGRAPHS) – AN EXAMPLE

Building and combining seeds

1. **circles 1, 2, and 3 processed independently**
2. **circles 2 and 3 synthesized yields circle 4**
3. **combination with circle 1 yields the complete answer**



PROCESSING WITH WEIGHTS

Motivation

Insufficient to propagate constraints about literal language use

Does not capture metonymic and metaphoric readings

Building combinations

Computing probabilities of local combinations

Choose the "best interpretation" for each reading of constrained items

Discard inferior local combinations for each of these interpretations

Combine local subgraphs and compute values for best combination

Example: **ACQUIRE - CORPORATION yields 0.9 (best)**
ACQUIRE - SOCIAL-EVENT yields 0.27 (discarded)
LEARN - CORPORATION yields 0.27 (worse alternative)
LEARN - SOCIAL-EVENT yields 0.081 (discarded)

BRANCH-AND-BOUND

The role of the other AI techniques

Constraint satisfaction for representation and problem partitioning

Solution synthesis for combining the answers

The role of branch-and-bound

Handles dependencies across subproblems

Provides estimates for the plausibility of combinations

In the example

***Adquirir* is the only word in circle 1 with dependencies outside the circle**

Correct word sense cannot be figured out within a single circle

For all possible meanings of *adquirir*,

the optimal meanings for the rest of the words can be determined

PROCESSING THE EXAMPLE (1)

CIRCLE 1: A, GR, DA

AFFECTED-VARS: A

POSSIBLE COMBINATIONS	SCORES A-GR		A-DA		
<A,acq>, <GR,org>, <DA,hum>	.9	*	.4	=	.36
<A,acq>, <GR,org>, <DA,org>	.9	*	1	=	.9
<A,learn>, <GR,org>, <DA,hum>	.8	*	.2	=	.16
<A,learn>, <GR,org>, <DA,org>	.8	*	.2	=	.16

Branch-and-Bound Reduction Output:

<A,acq>, <GR,org>, <DA,org>	.9
<A,learn>, <GR,org>, <DA,hum>	.16

PROCESSING THE EXAMPLE (2)

CIRCLE 2: A, C, ATD

AFFECTED-VARS: A, C

POSSIBLE COMBINATIONS	SCORES
<A,acq>, <C,corp>, <ATD,loc>	.8
<A,acq>, <C,corp>, <ATD,instr>	.9
<A,acq>, <C,event>, <ATD,loc>	.24
<A,acq>, <C,event>, <ATD,instr>	.27
<A,learn>, <C,corp>, <ATD,loc>	.24
<A,learn>, <C,corp>, <ATD,instr>	.27
<A,learn>, <C,event>, <ATD,loc>	.24
<A,learn>, <C,event>, <ATD,instr>	.27

Branch-and-Bound Reduction Output:

<A,acq>, <C,corp>, <ATD,instr>	.9
<A,acq>, <C,event>, <ATD,instr>	.27
<A,learn>, <C,corp>, <ATD,instr>	.27
<A,learn>, <C,event>, <ATD,instr>	.27

PROCESSING THE EXAMPLE (3)

SYNTHESIS CIRCLES 2 and 3 to create CIRCLE 4

AFFECTED-VARS: A

<A,acq>, <C,corp>, <ATD,instr>	**.9**	
<A,acq>, <C,event>, <ATD,instr>	**.27**	
<A,learn>, <C,corp>, <ATD,instr>	**.27**	
<A,learn>, <C,event>, <ATD,instr>	**.27**	**plus**
<C,corp>, <E,loc>, <ESP,nat>	**1.0**	
<C,event>, <E,loc>, <ESP,nat>	**1.0**	**yields the possible combinations**

<A,acq>, <C,corp>, <ATD,instr>, <E,loc>, <ESP,nat>,<S,own>	**.9**
<A,acq>, <C,event>, <ATD,instr>, <E,loc>, <ESP,nat>,<S,own>	**.27**
<A,learn>, <C,corp>, <ATD,instr>, <E,loc>, <ESP,nat>,<S,own>	**.27**
<A,learn>, <C,event>, <ATD,instr>, <E,loc>, <ESP,nat>,<S,own>	**.27**

Branch-and-Bound Reduction Output:

<A,acq>, <C,corp>, <ATD,instr>, <E,loc>, <ESP,nat>,<S,own>	**.9**
<A,learn>, <C,corp>, <ATD,instr>, <E,loc>, <ESP,nat>,<S,own>	**.27**

PROCESSING THE EXAMPLE (4)

SYNTHESIS CIRCLES 1 and 4 to create CIRCLE 5
AFFECTED-VARS: none

<A,acq>, <GR,org>, <DA,org>	**.9**
<A,learn>, <GR,org>, <DA,hum>	**.16**

plus

<A,acq>, <C,corp>, <ATD,instr>, <E,loc>, <ESP,nat>,<S,own>	**.9**
<A,learn>, <C,corp>, <ATD,instr>, <E,loc>, <ESP,nat>,<S,own>	**.27**

yields the possible combinations

<A,acq>, <GR,org>, <DA,org>,<C,corp>, <ATD,instr>, <E,loc>, <ESP,nat>,<S,own>	**.81**
<A,acq>, <GR,org>, <DA,org>,<C,corp>, <ATD,instr>, <E,loc>, <ESP,nat>,<S,own>	**.04**

Branch-and-Bound Reduction Output:

<A,acq>, <GR,org>, <DA,org>,<C,corp>, <ATD,instr>, <E,loc>, <ESP,nat>,<S,own>	**.81**

RESULTS IN SEMANTIC ANALYSIS

Text	*Roche*	*Reality-Refund*	*Matra*	*Comercio Brasilieno*	*Average*
#words	**347**	**385**	**370**	**353**	**364**
#sentences	**21**	**16**	**14**	**17**	**17**
words/sentence	**16.5**	**24.0**	**26.4**	**20.8**	**21.4**
#open-class	**183**	**167**	**177**	**177**	**176**
#ambiguous	**57**	**42**	**57**	**35**	**48**
#resolved by syntax	**21**	**19**	**20**	**12**	**18**
#ambiguous after syntax	**36**	**23**	**37**	**23**	**30**
#correctly resolved	**30**	**22**	**25**	**22**	**25**
#ambiguous correct	**89%**	**98%**	**79%**	**97%**	**91%**
#correct overall	**97%**	**99%**	**93%**	**99%**	**97%**

COMPLEXITY RESULTS

$O(n\, p^c)$ "near linear time"

- **n = number of circles, proportional to length of input**
- **p = maximum number answers after branch-and-bound reduction for a circle**
- **c = maximum number of input circles for a single circle**

c is normally 2 or 3 for NLP

p is kept low (tree-shaped input results in only 1 affected variable per circle)

occasional long-distance dependencies cause delays in optimization

(responsible for non-linear effects)

Sample sentences	***A***	***B***	***C***
#plans	**79**	**95**	**119**
exhaustive combinations	**7,864,320**	**56,687,040**	**235 billion**
hunter gatherer	**179**	**254**	**327**