

# Recent advances in lifted inference @ Leuven

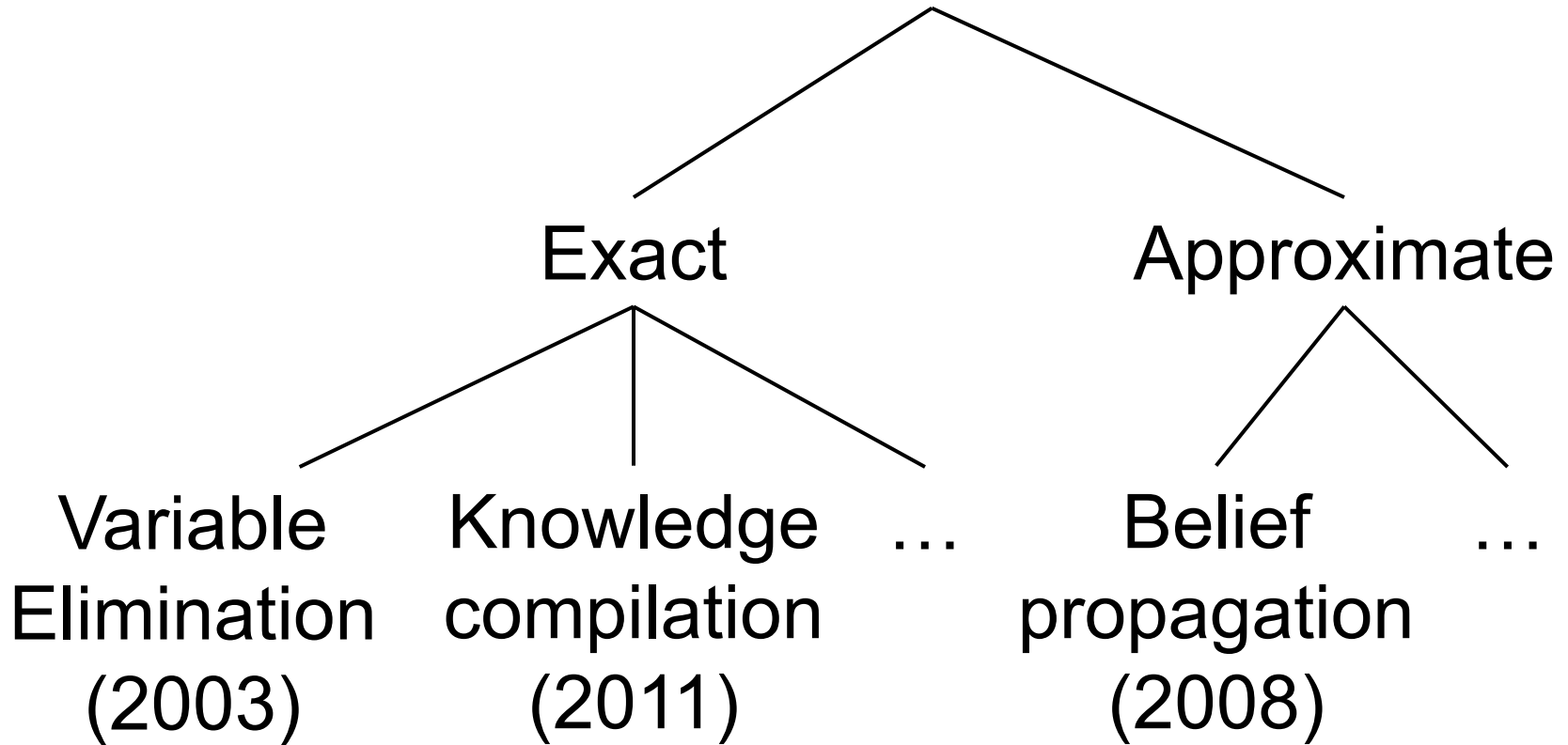
H. Blockeel, J. Davis, L. De Raedt, D. Fierens,  
W. Meert, N. Taghipour, G. Van den Broeck

*SML, April 19, 2012*

# Outline

- Introduction to lifted inference
- Four contributions
  - Arbitrary constraints
  - Completeness results
  - Conditioning
  - An approximate method

# Lifted inference



**and many more !**

**MLN**

1.5 Attends(person) → Series

1.2 Topic → Attends(person)

**MLN**

1.5 Attends(person)  $\rightarrow$  Series

1.2 Topic  $\rightarrow$  Attends(person)

Series

Attends( $p_1$ )

Attends( $p_2$ )

...

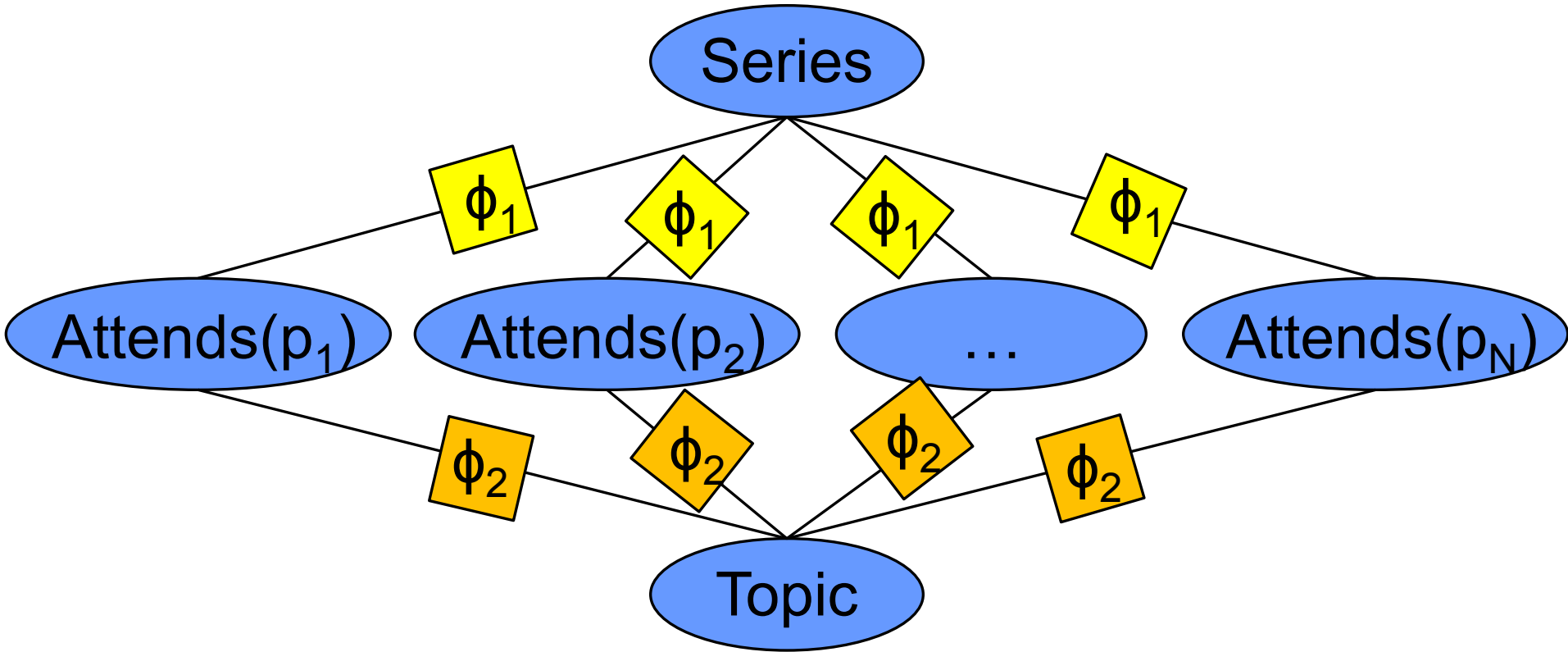
Attends( $p_N$ )

Topic

MLN

1.5 Attends(person)  $\rightarrow$  Series

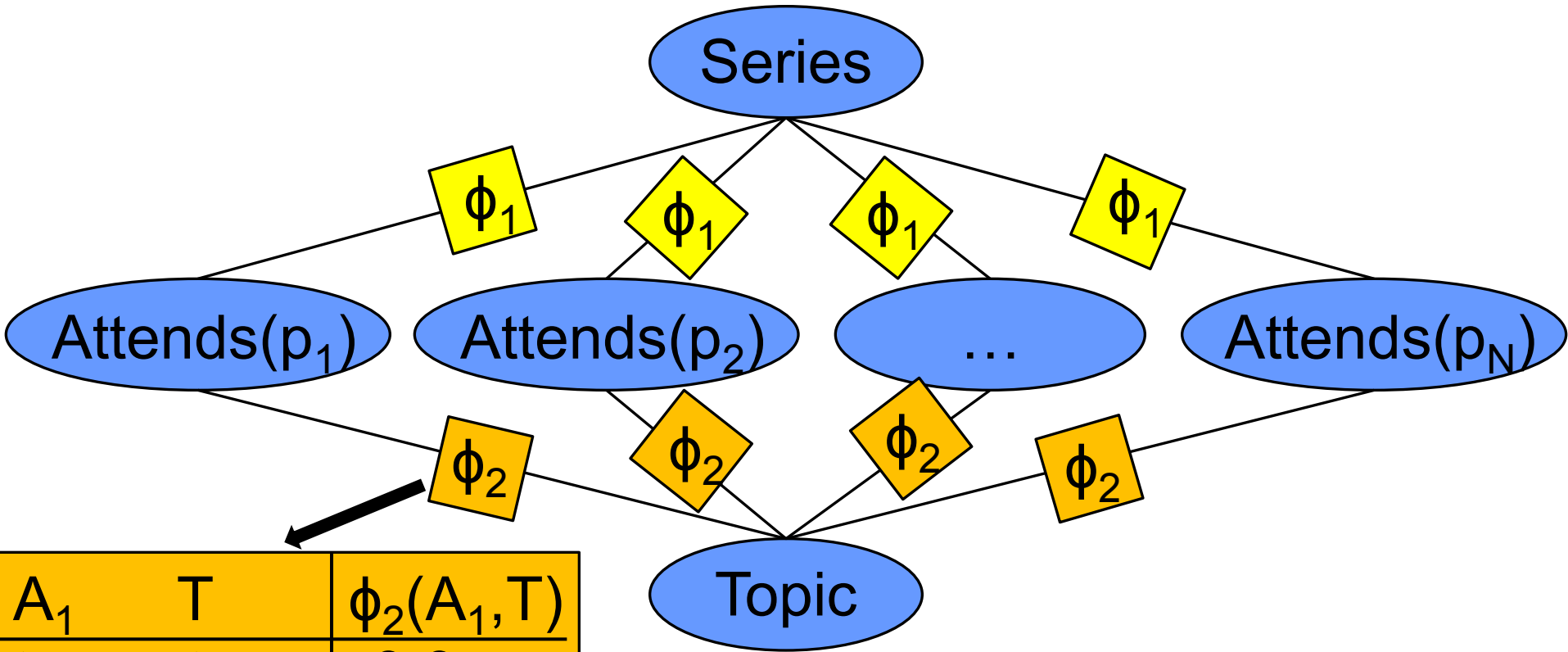
1.2 Topic  $\rightarrow$  Attends(person)



MLN

1.5 Attends(person)  $\rightarrow$  Series

1.2 Topic  $\rightarrow$  Attends(person)

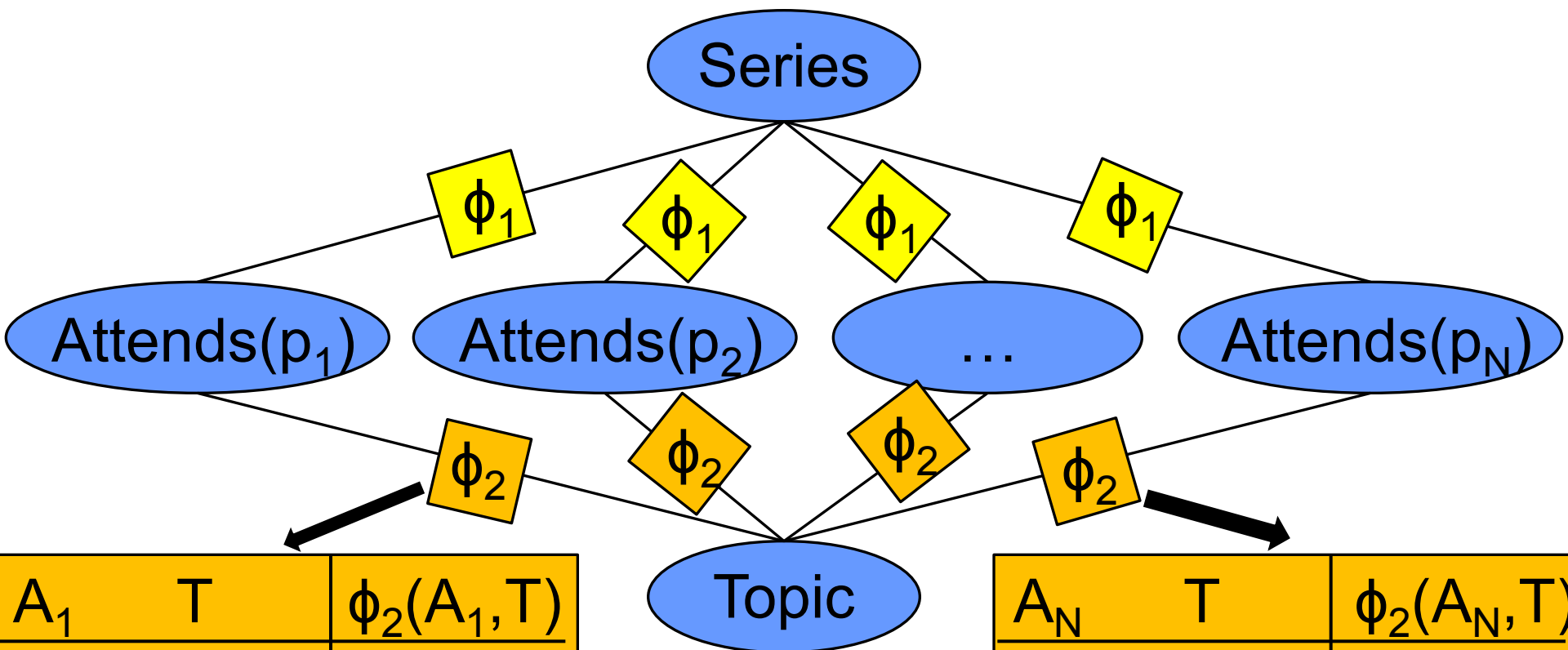


A <sub>1</sub>	T	$\phi_2(A_1, T)$
true	true	3.3
true	false	3.3
false	true	1.0
false	false	3.3

MLN

1.5 Attends(person)  $\rightarrow$  Series

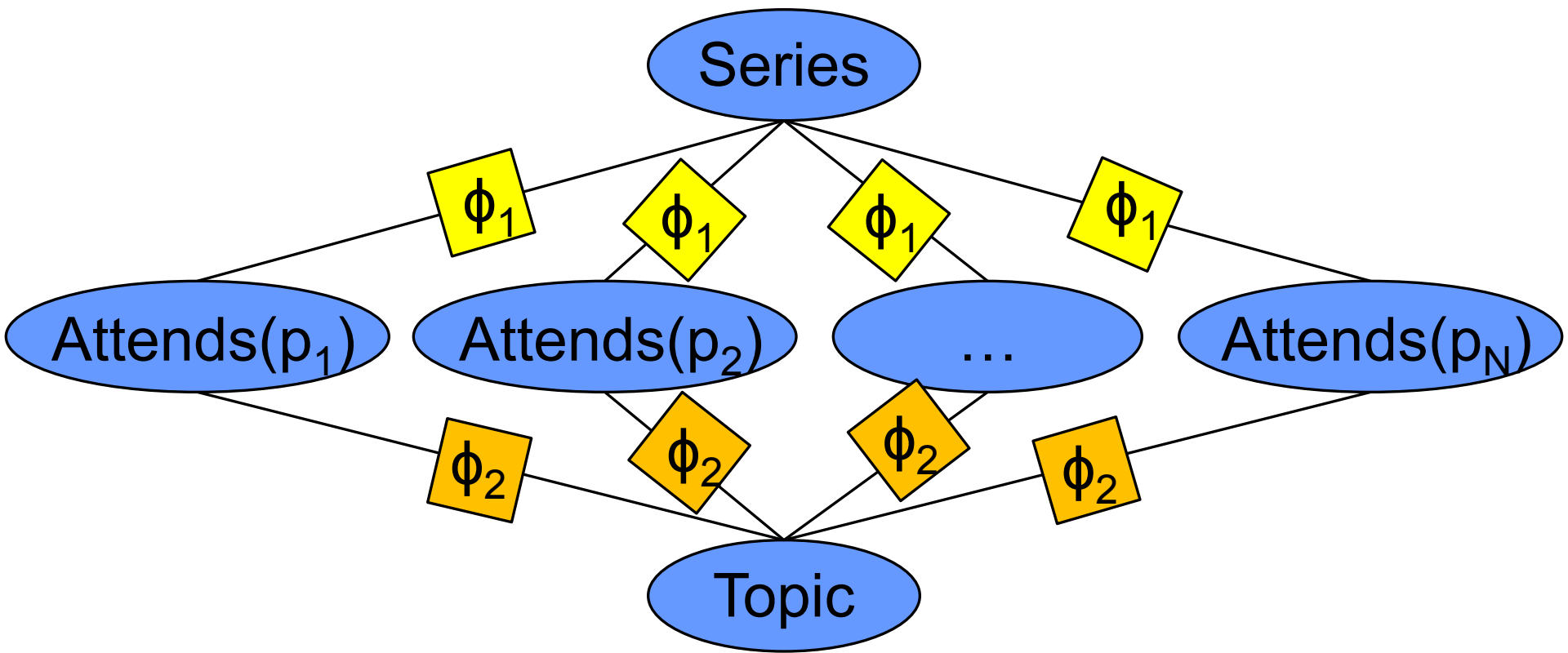
1.2 Topic  $\rightarrow$  Attends(person)

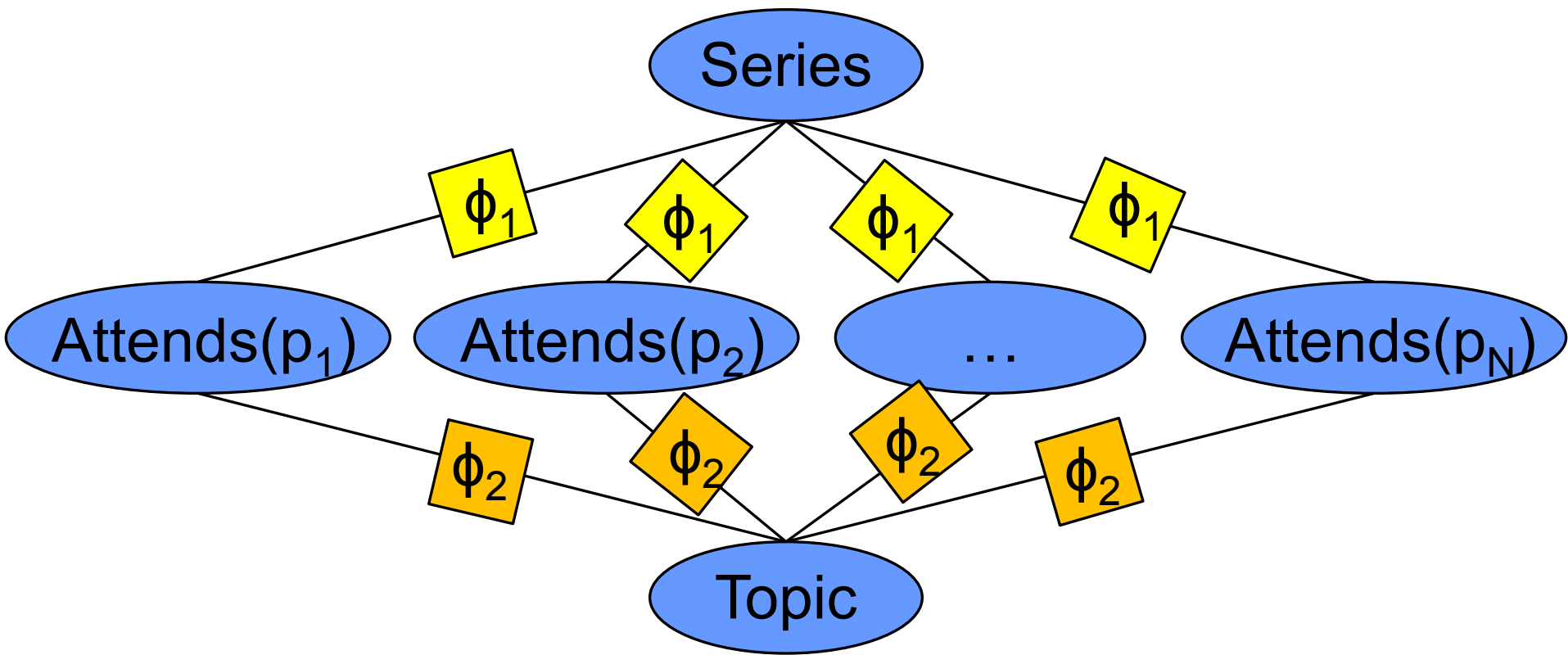


$A_1$	T	$\phi_2(A_1, T)$
true	true	3.3
true	false	3.3
false	true	1.0
false	false	3.3

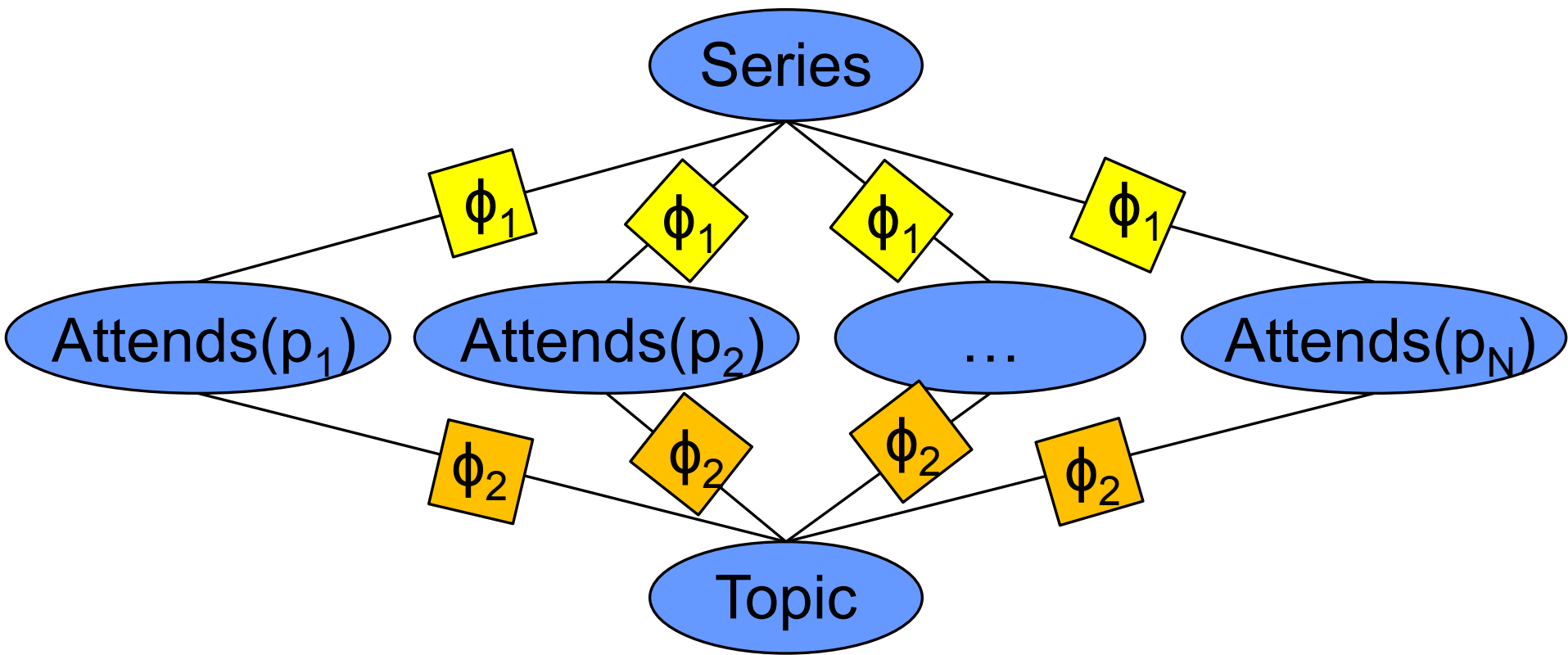
$A_N$	T	$\phi_2(A_N, T)$
true	true	3.3
true	false	3.3
false	true	1.0
false	false	3.3





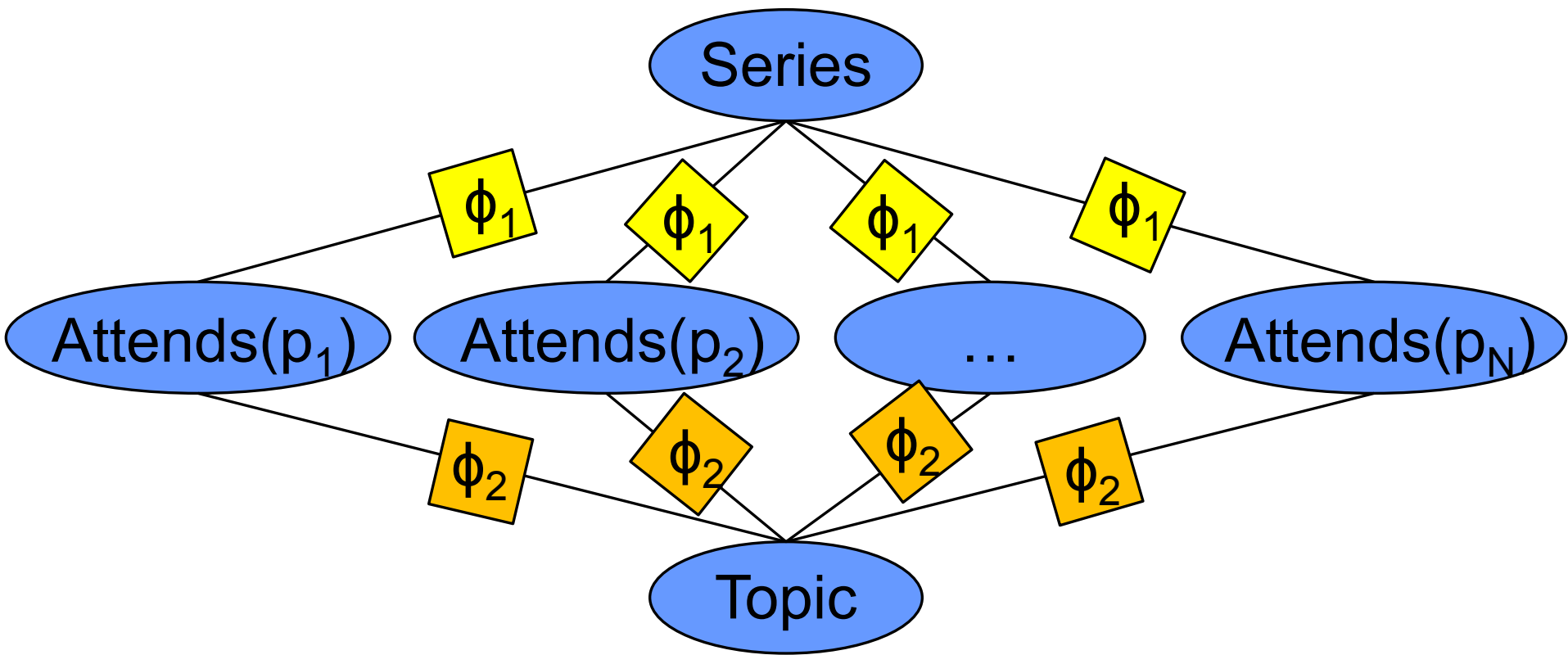


$$P(S, A_1, \dots, A_N, T) = \frac{1}{Z} \prod_{i=1}^N \phi_1(A_i, S) \prod_{i=1}^N \phi_2(T, A_i)$$



$$P(S) = \frac{1}{Z} \sum_T \sum_{A_1} \dots \sum_{A_N} \prod_{i=1}^N \phi_1(A_i, S) \prod_{i=1}^N \phi_2(T, A_i)$$

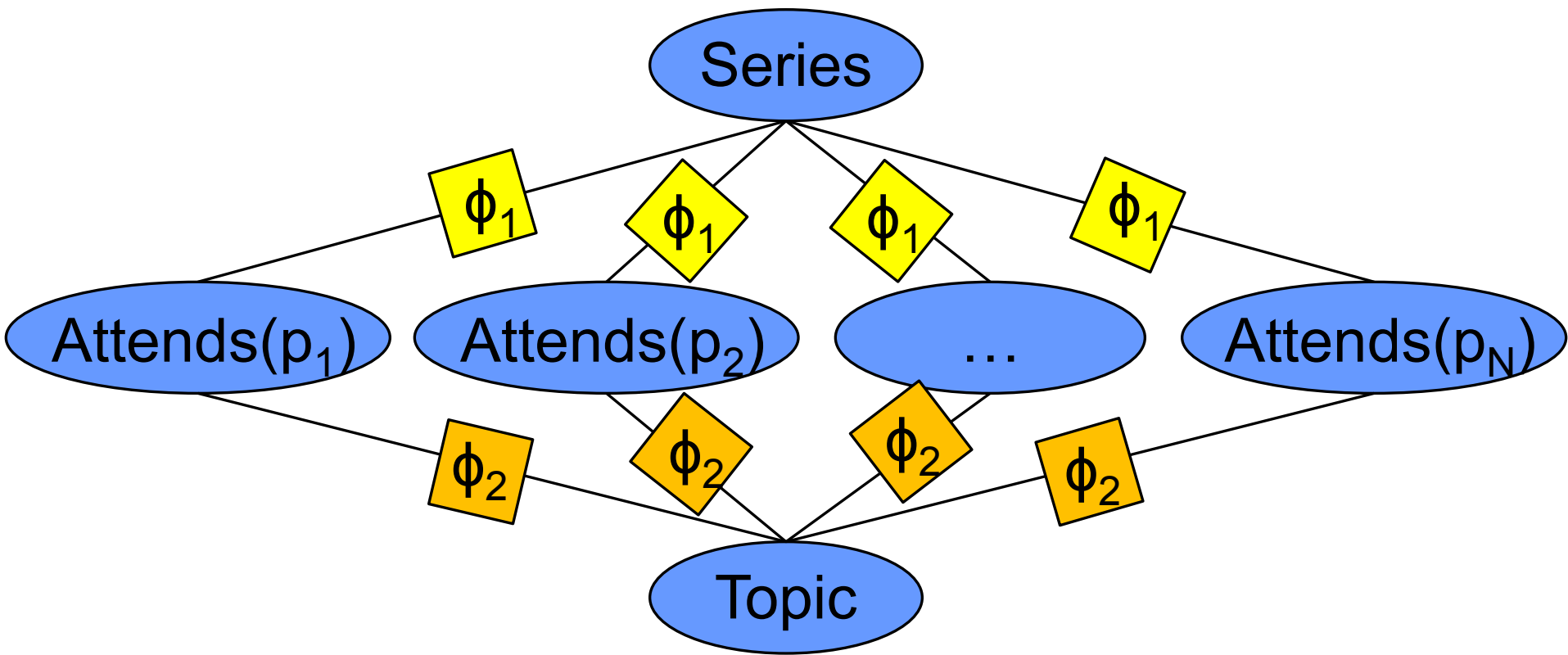
will it become  
a series ?



$$P(S) = \frac{1}{Z} \sum_T \sum_{A_1} \dots \sum_{A_N} \prod_{i=1}^N \phi_1(A_i, S) \prod_{i=1}^N \phi_2(T, A_i)$$

will it become  
a series ?

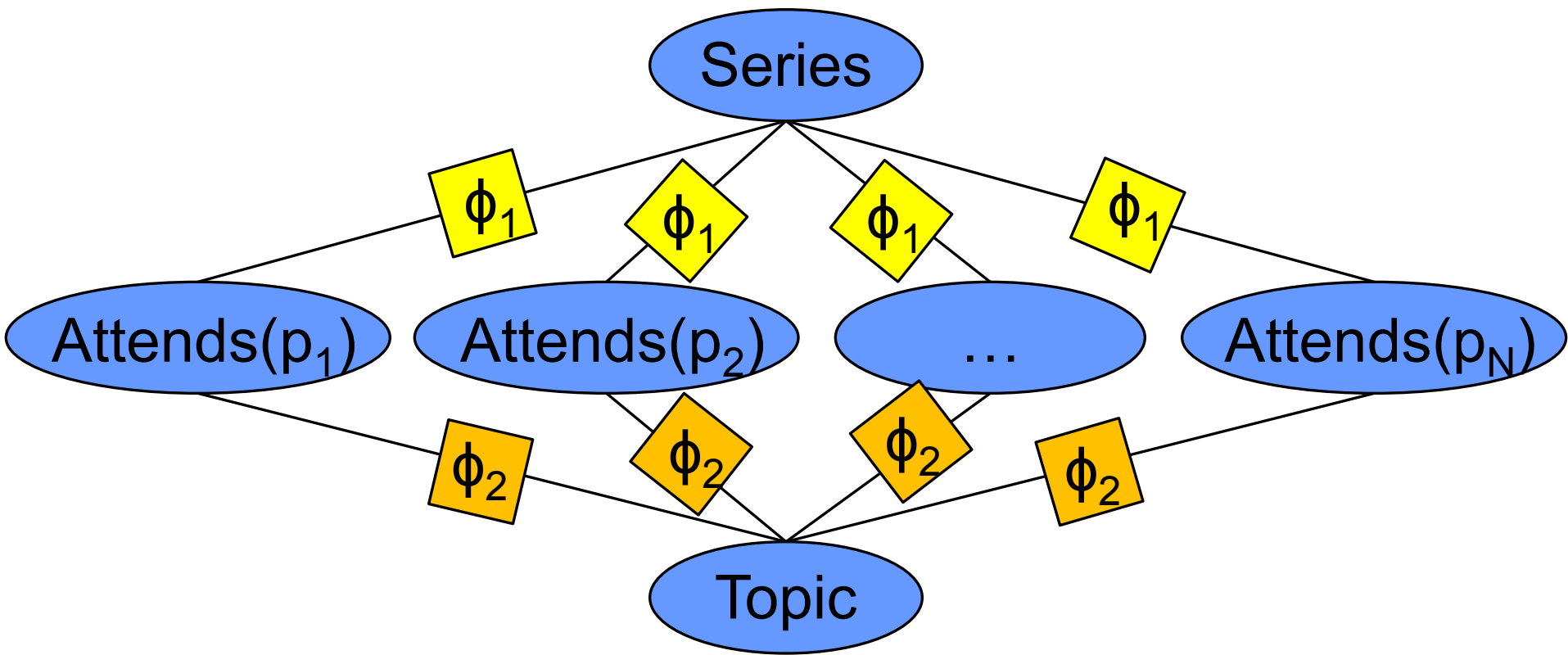
$2^{(N+1)}$  terms



$$\sum_T \sum_{A_1} \dots \sum_{A_N} \prod_{i=1}^N \phi_1(A_i, S) \prod_{i=1}^N \phi_2(T, A_i)$$

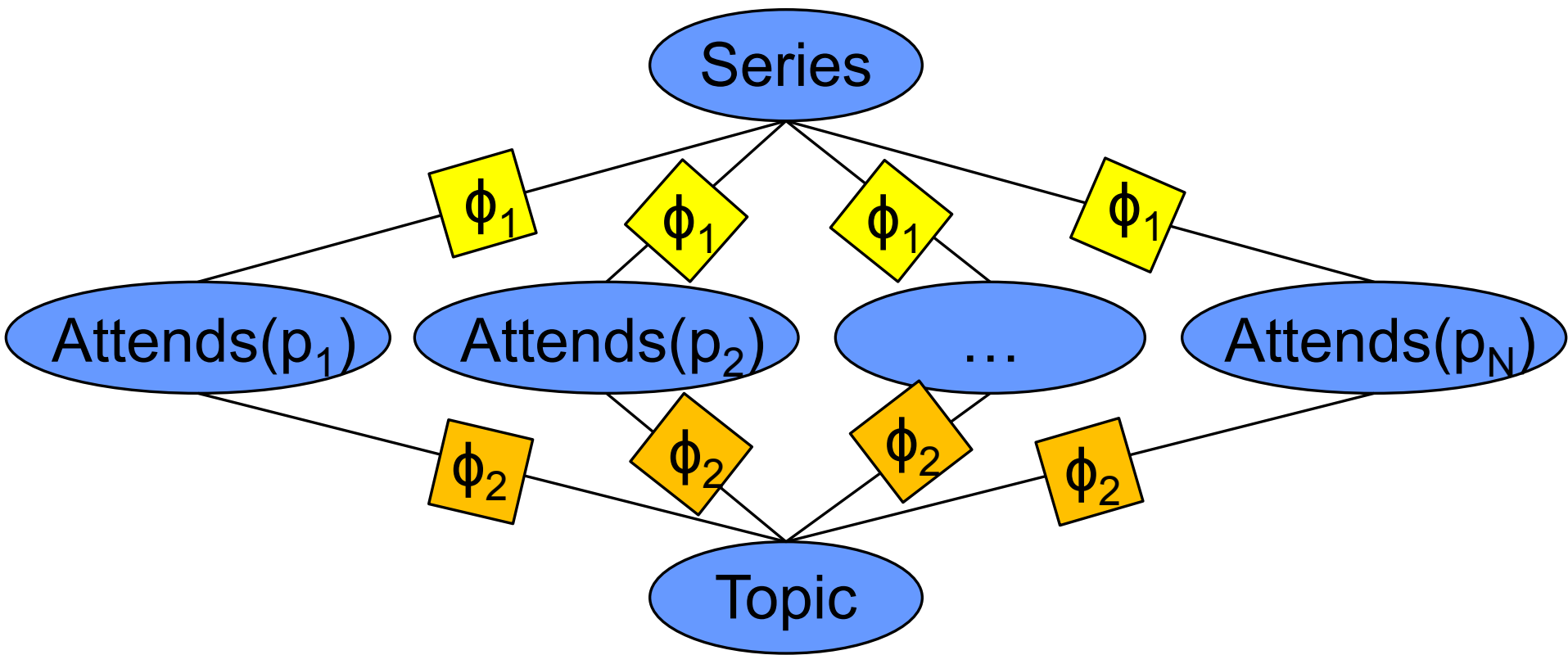


**$2^{(N+1)}$  terms**



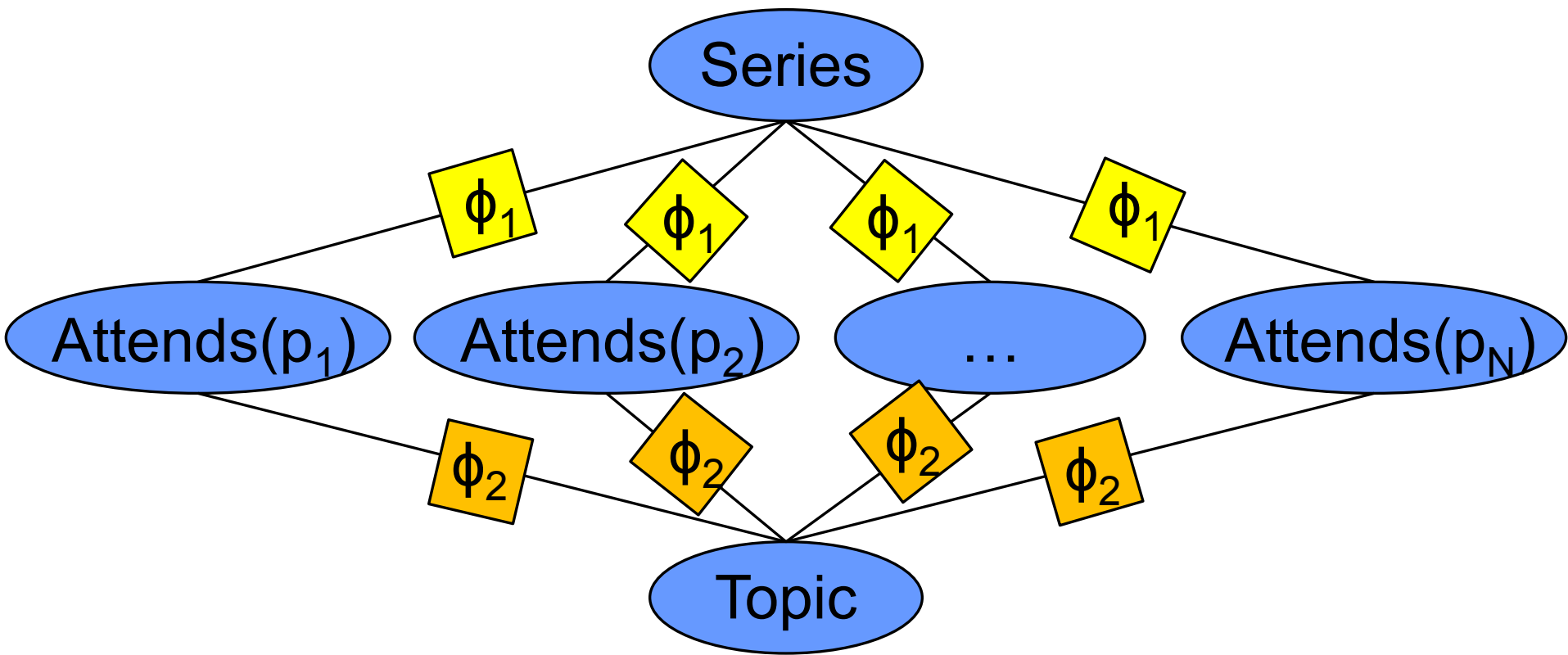
$$\sum_T \left( \sum_{A_1} \phi_1(A_1, S) \phi_2(T, A_1) \right) \dots \left( \sum_{A_N} \phi_1(A_N, S) \phi_2(T, A_N) \right)$$

1 for every person



$$\sum_T \left( \sum_{A_1} \underbrace{\phi_1(A_1, S) \phi_2(T, A_1)} \right) \dots \left( \sum_{A_N} \underbrace{\phi_1(A_N, S) \phi_2(T, A_N)} \right)$$

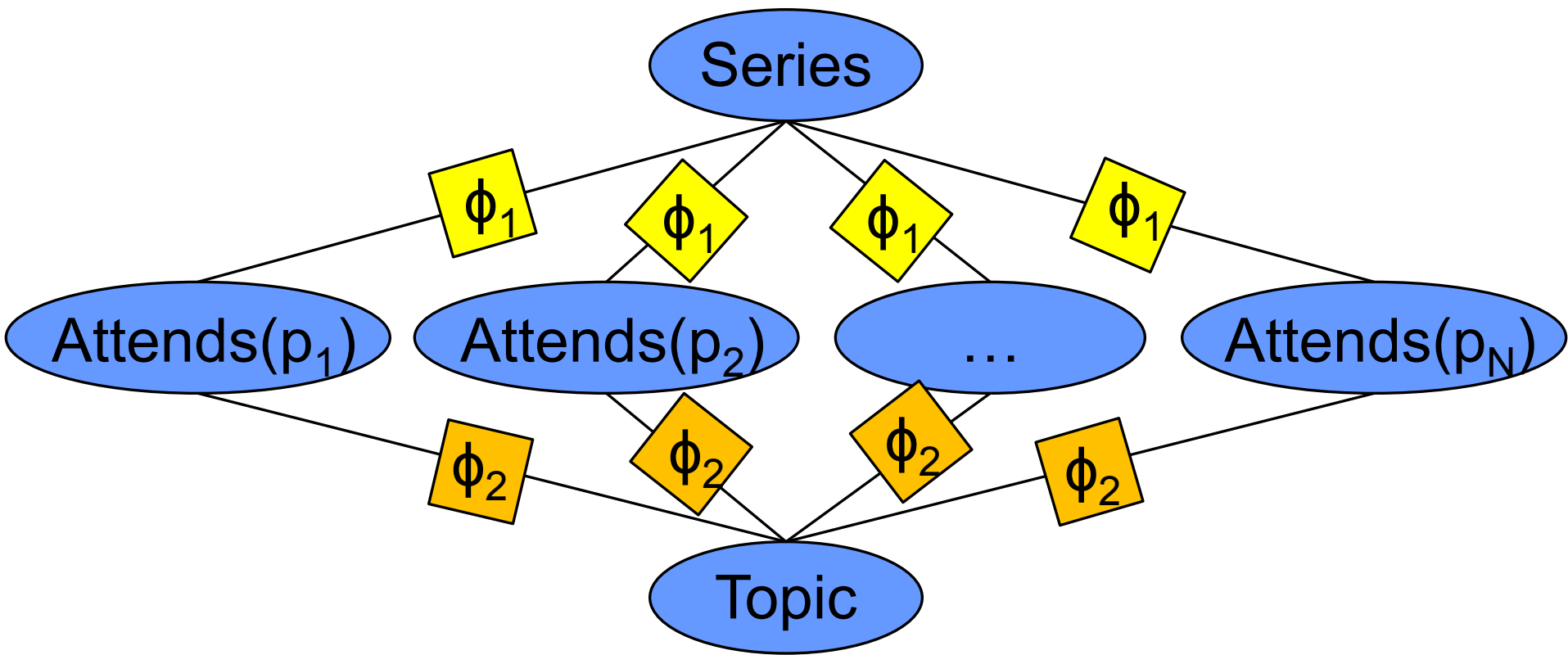
**N times the same product !**



$$\sum_T \left( \underbrace{\sum_{A_1} \phi_1(A_1, S) \phi_2(T, A_1)} \right) \dots \left( \underbrace{\sum_{A_N} \phi_1(A_N, S) \phi_2(T, A_N)} \right)$$

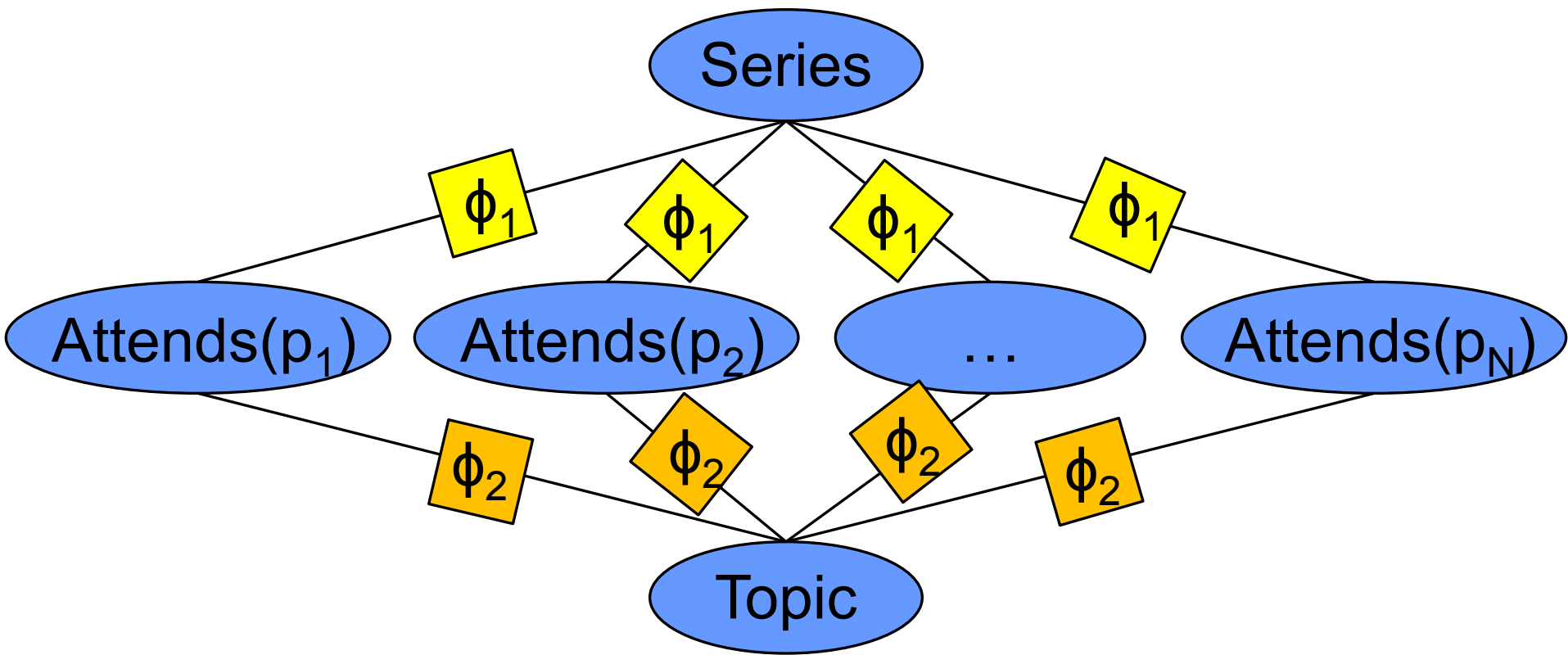
**N times the same sum !**





lifted: 
$$\sum_T \left( \underbrace{\sum_A \phi_1(A, S) \phi_2(T, A)} \right)^N$$

**compute only once !**



lifted: 
$$\sum_T \left( \sum_A \underbrace{\phi_1(A, S)}_{\text{"lifted sum-out"}} \underbrace{\phi_2(T, A)}_{\text{"lifted multiplication"}} \right)^N$$

**"lifted sum-out"**

**"lifted multiplication"**

# Lifted Variable Elimination

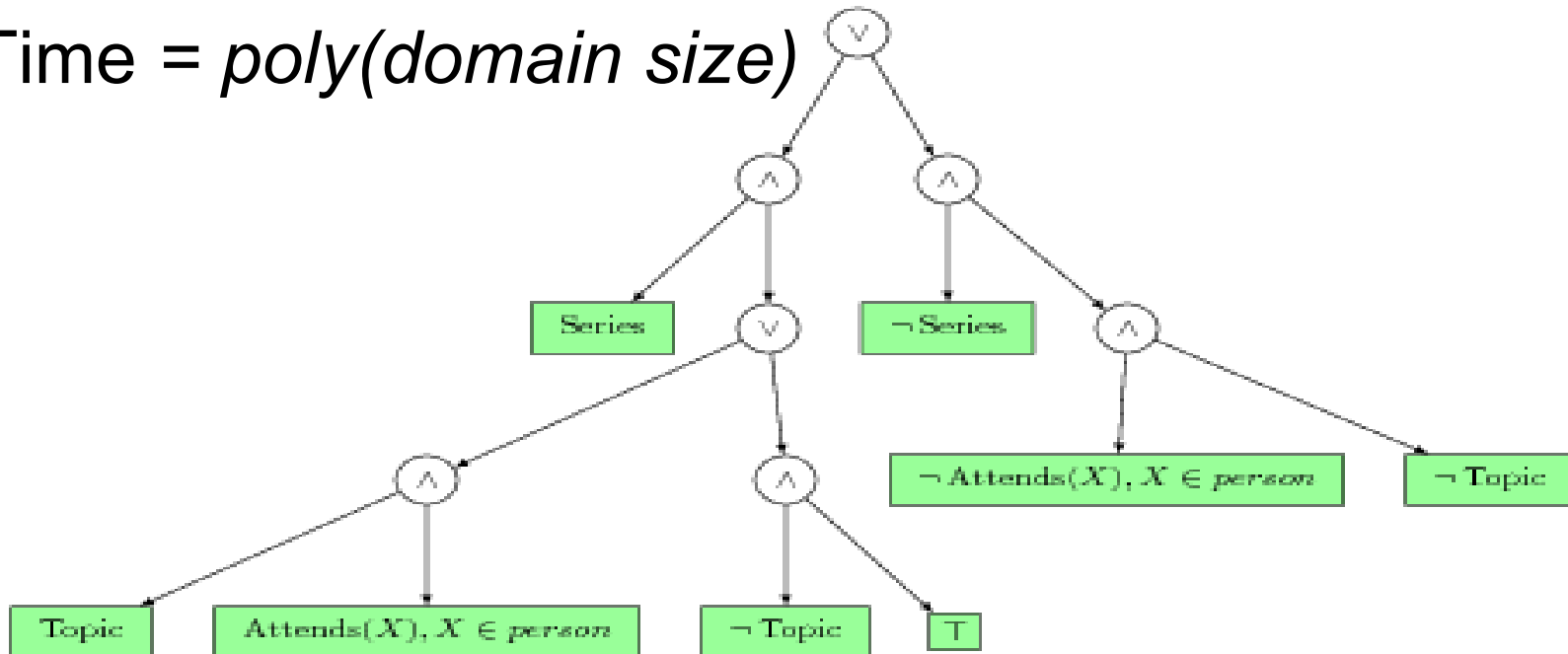
[Poole '03,...]

- Repeatedly apply certain **operators** on the model
  - Lifted multiplication
  - Lifted sum-out
  - ...
- Until the desired result is found

# Lifted Knowledge Compilation

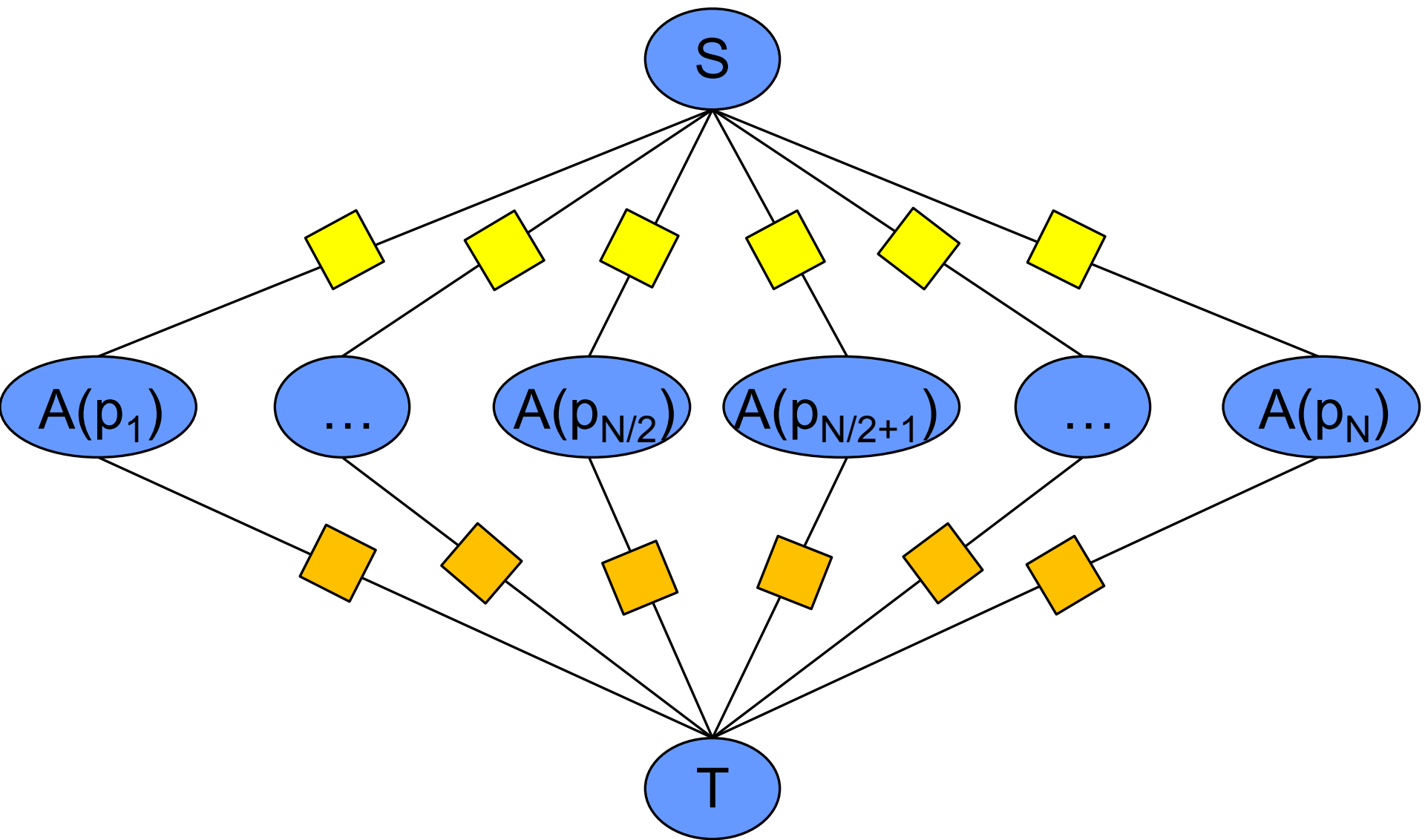
[Van den Broeck et al '11,...]

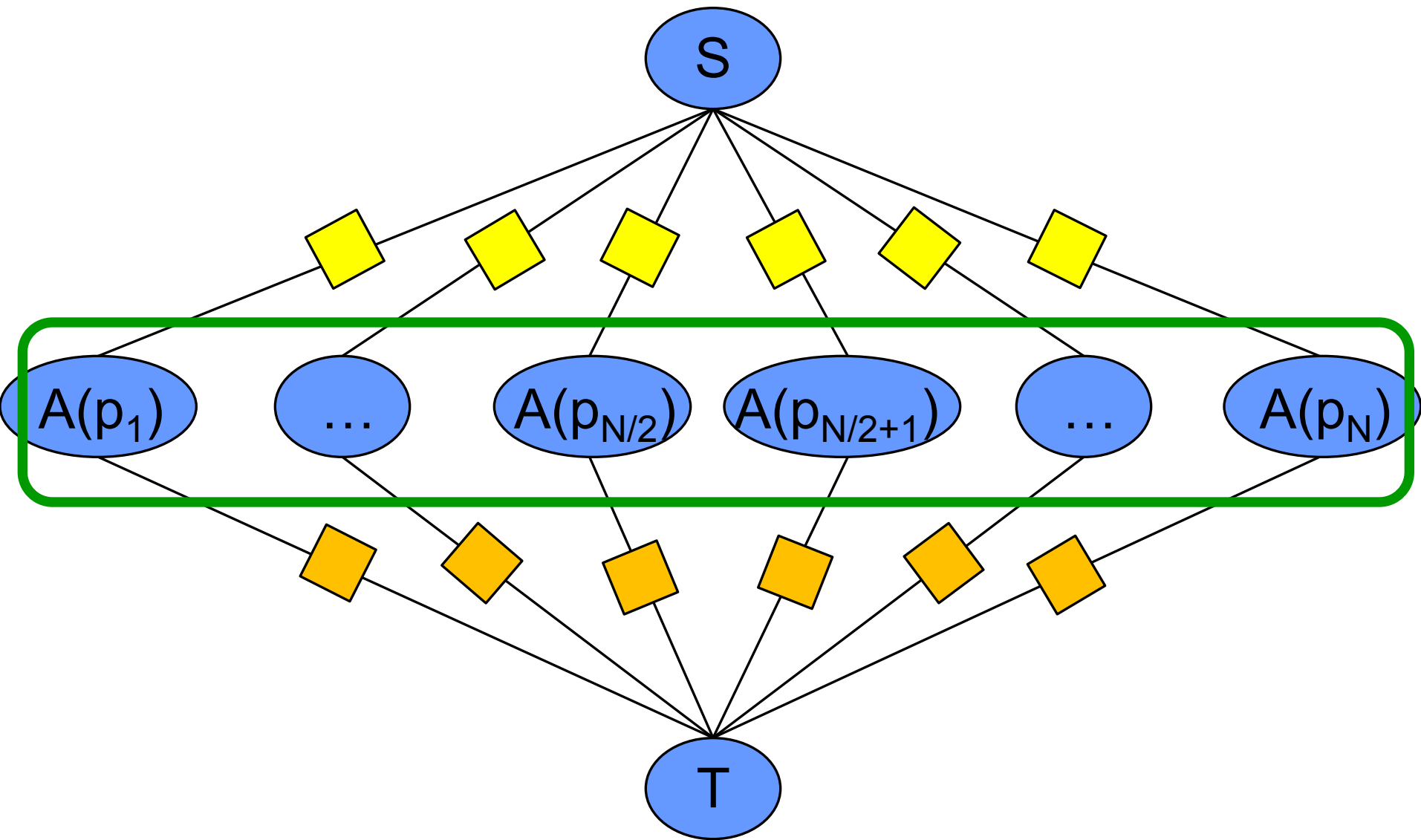
- Compile the model into a “**lifted**” circuit (“FO d-DNNF”)
  - How? Compilation rules
- Inference = traversing the circuit
  - Time = *poly(domain size)*

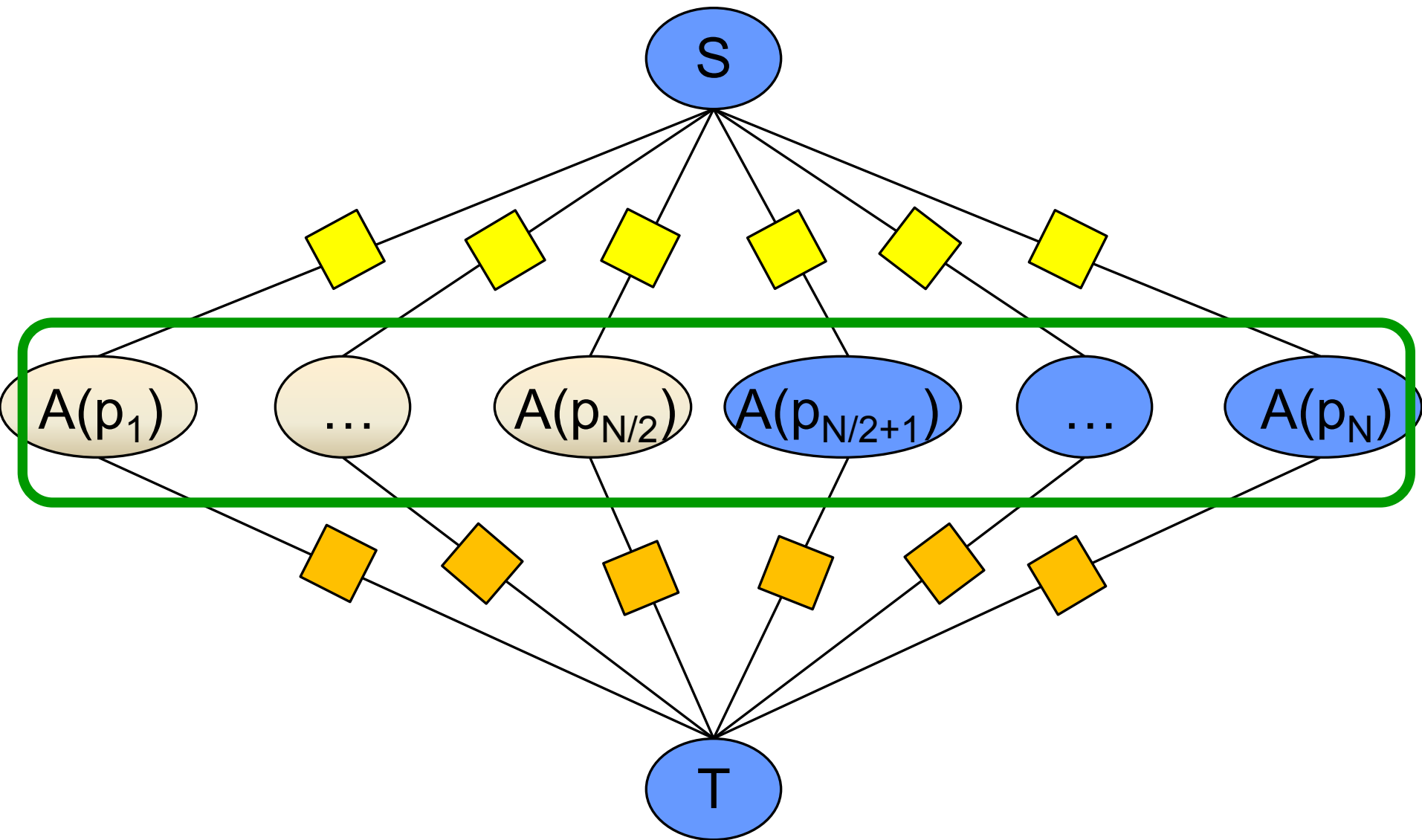


# Outline

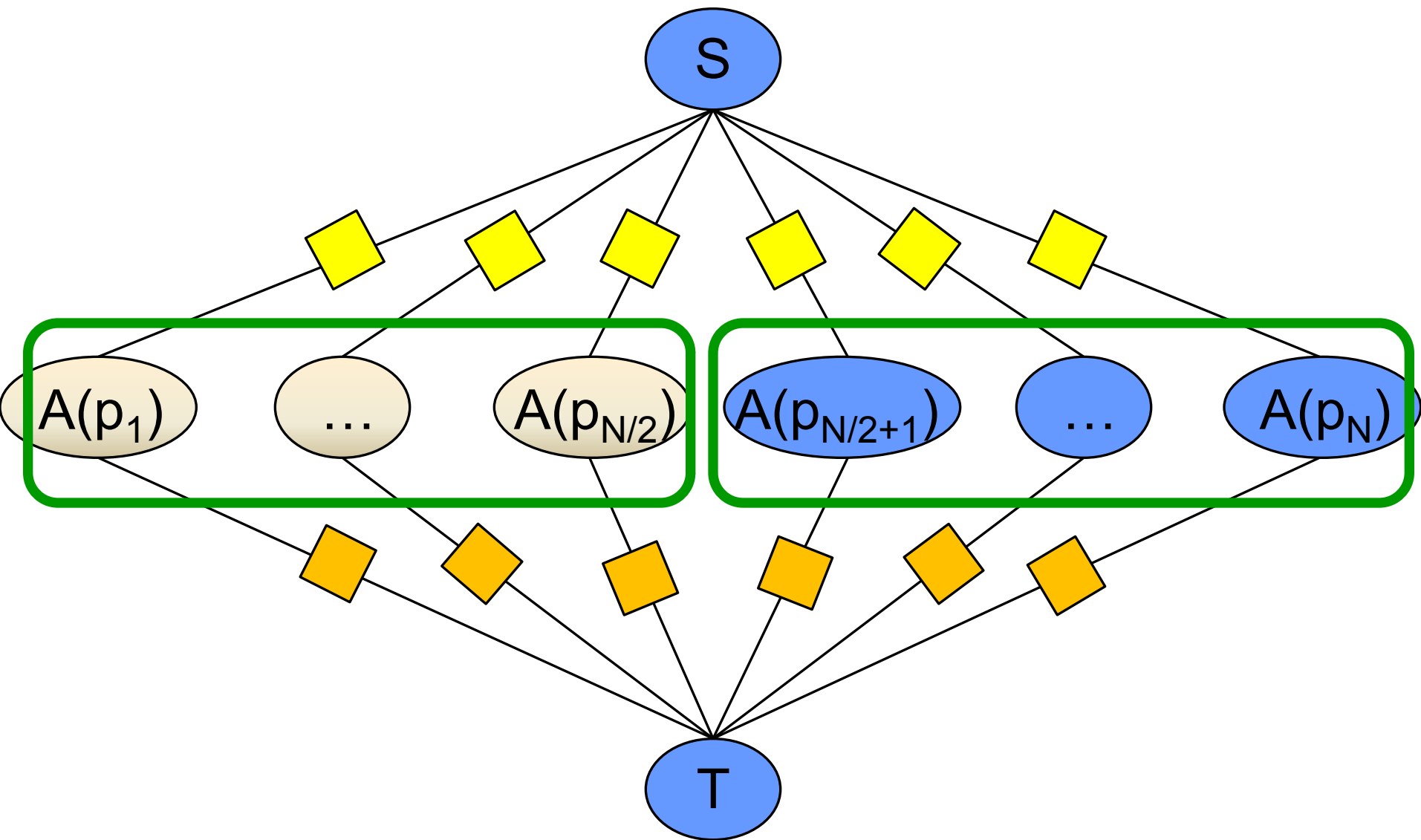
- Introduction to lifted inference
- Four contributions
  - Arbitrary constraints
  - Completeness results
  - Conditioning
  - An approximate method

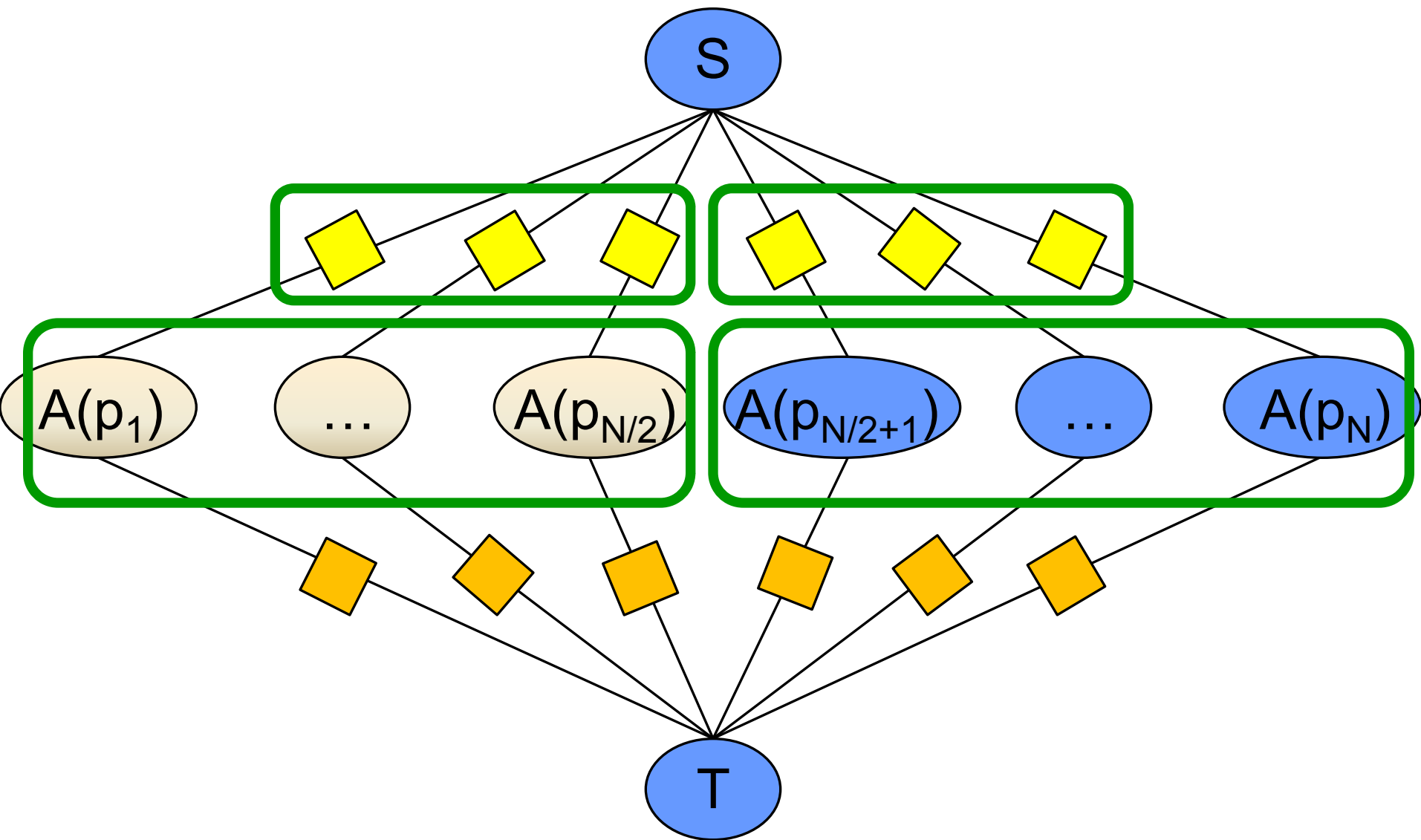


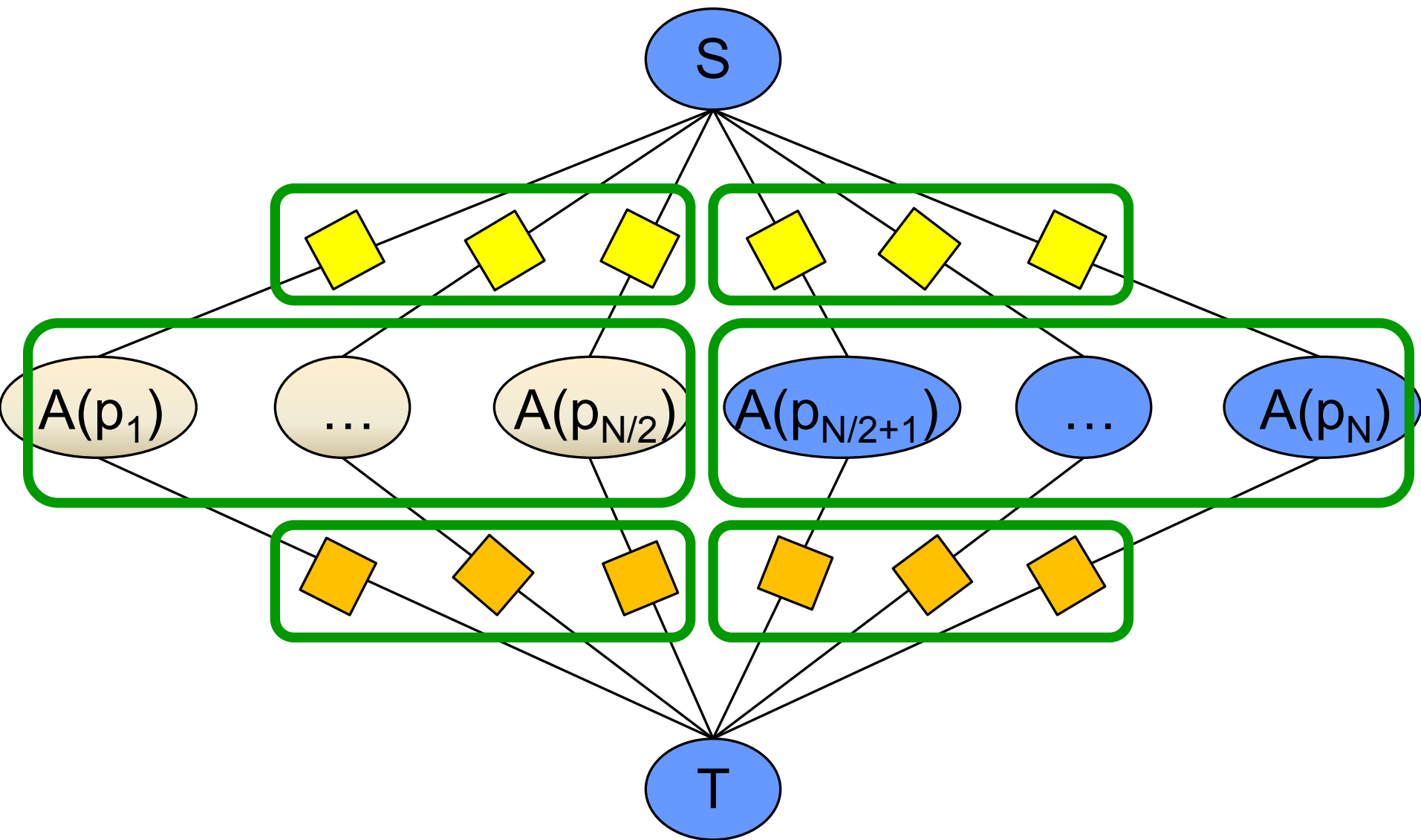


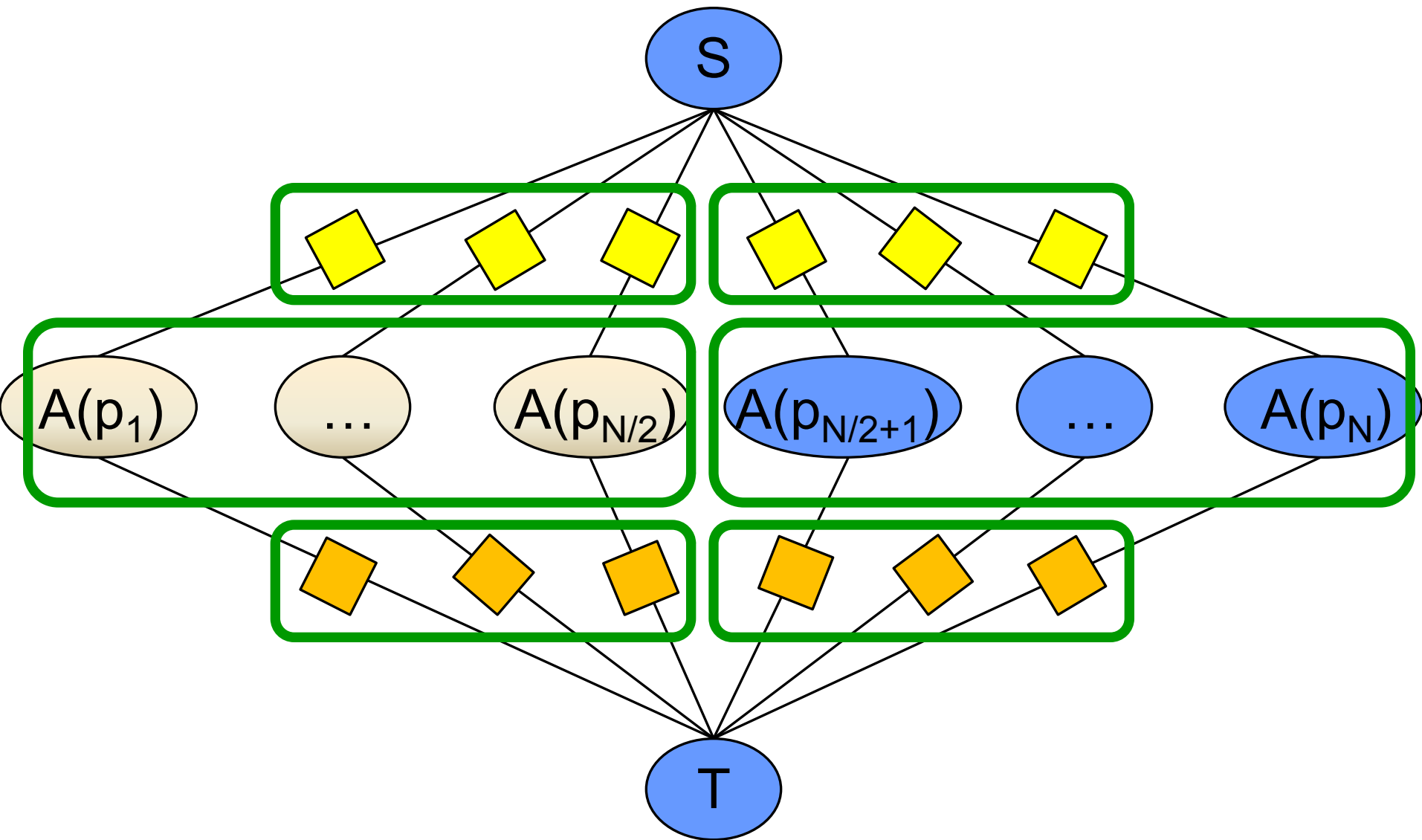




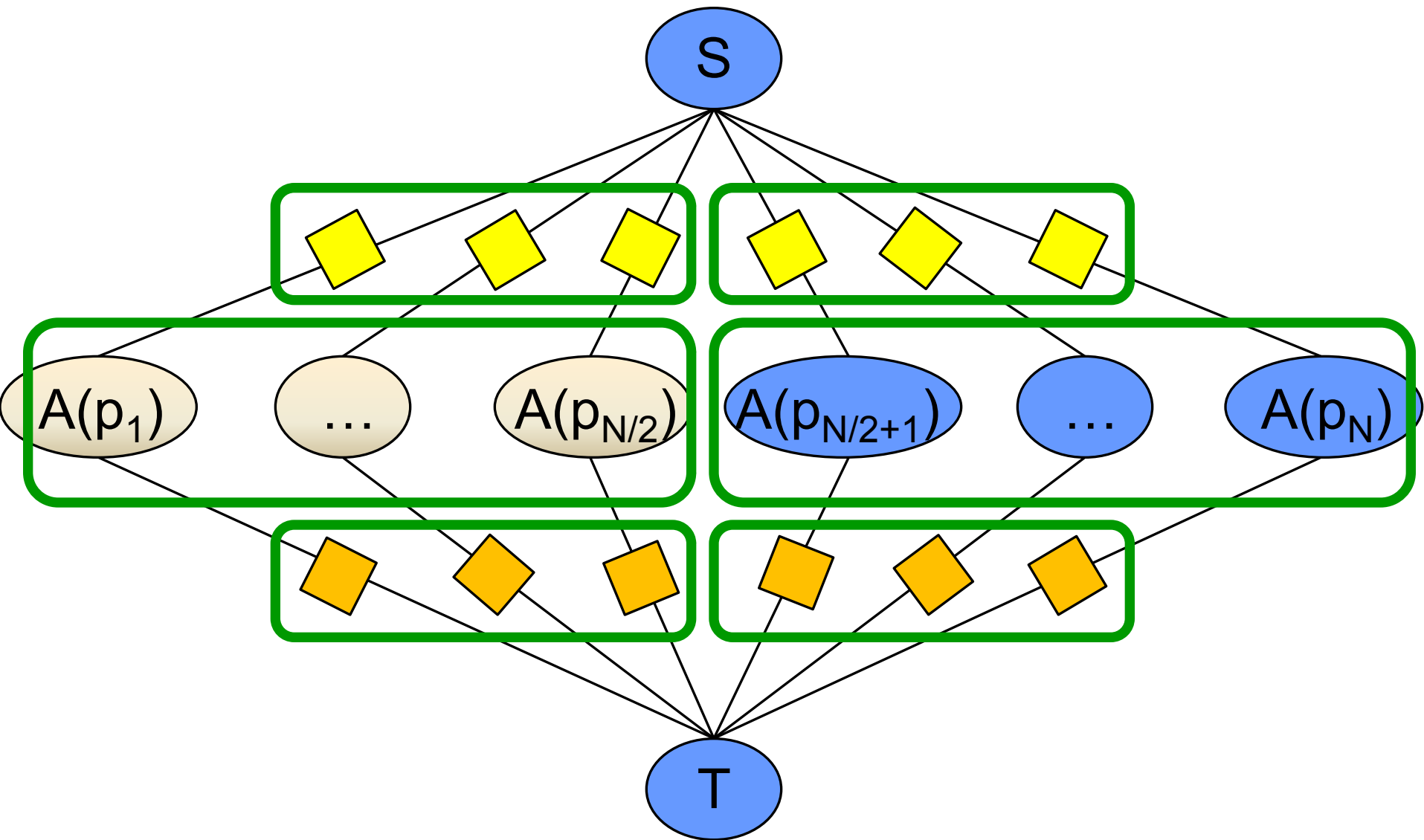








Bigger groups = more lifting !



Bigger groups = more lifting !

The groups are specified by **constraints**

# Importance of constraints

[Taghipour et al, AISTATS'12]

- Exact lifted algorithms use a particular constraint language

group  $\rightarrow$  constraint  $\rightarrow$  can it be expressed  
in the language ?

- Often leads to unnecessarily small groups  
 $\rightarrow$  less lifting

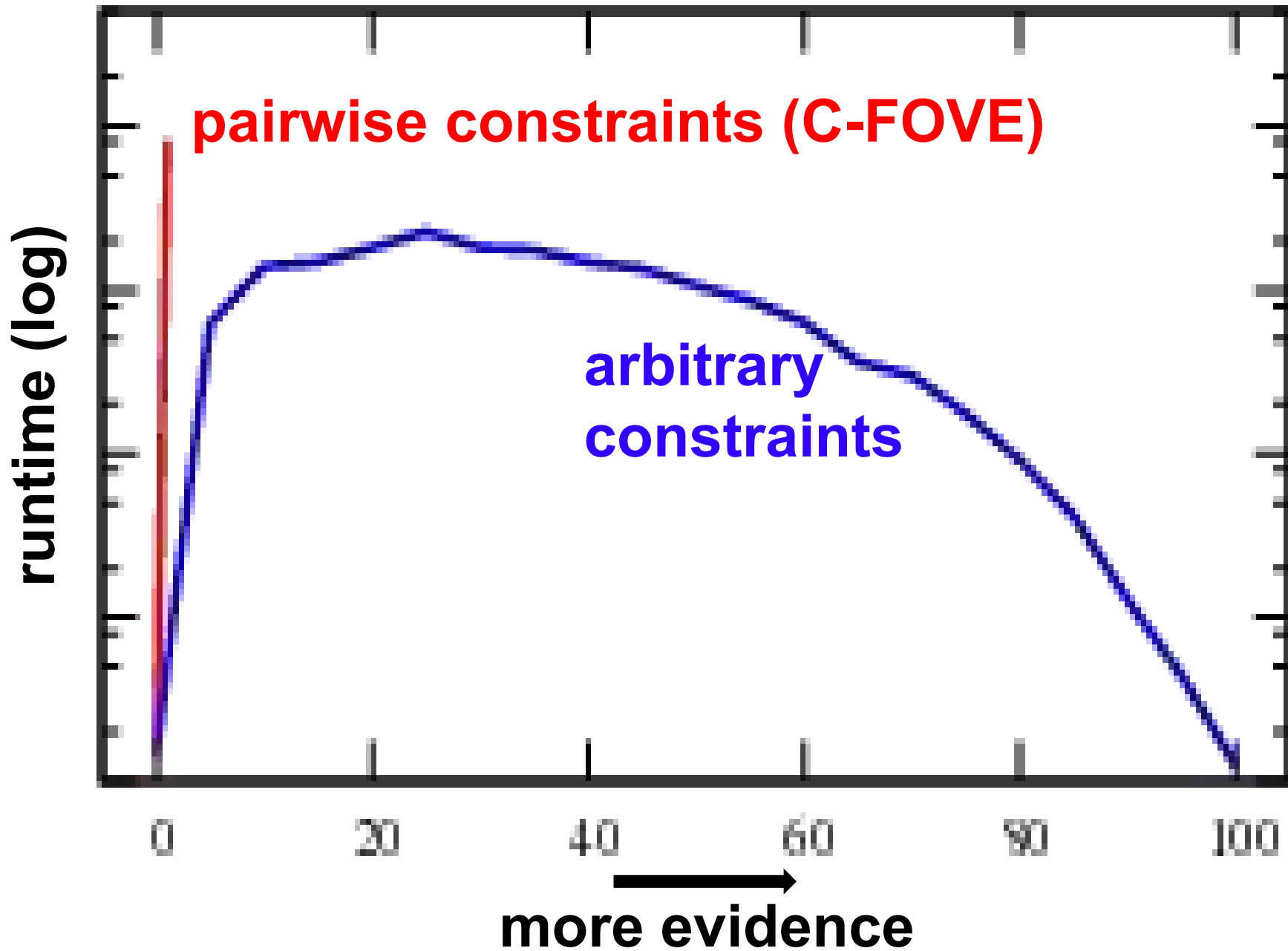
# Importance of constraints

[Taghipour et al, AISTATS'12]

- Exact lifted algorithms use a particular constraint language

group  $\rightarrow$  constraint  $\rightarrow$  can it be expressed  
in the language ?

- Often leads to unnecessarily small groups  
 $\rightarrow$  less lifting
- We avoid using a particular constraint language  
Instead: **arbitrary constraints**  
+ relational algebra





# Outline

- Introduction to lifted inference
- Four contributions
  - Arbitrary constraints
  - Completeness results
  - Conditioning
  - An approximate method

# Outline

- Introduction to lifted inference
- Four contributions
  - Arbitrary constraints
  - Completeness result
  - Conditioning
  - Approximate inference

# Outline

- Introduction to lifted inference
- Four contributions
  - Arbitrary constraints
  - **Completeness result**
  - Conditioning
  - Approximate inference

# What is Lifted Inference?

- Propositional inference is **intractable**

Solution: **lifted inference**

*“Exploit symmetries”*

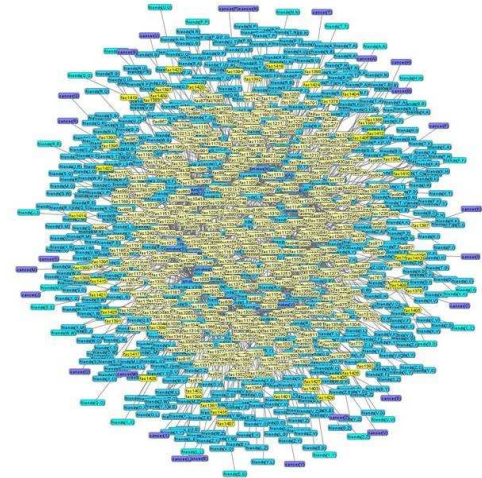
*“Reason at first-order level”*

*“Reason about groups of objects as a whole”*

*“Avoid repeated computations”*

*“Mimic resolution in theorem proving”*

- There is a common understanding but **no formal definition** of lifted inference!



# What is Lifted Inference?

- What is commonly understood as exact lifted inference?

## Definition: Domain-Lifted Inference

Complexity of computing  $P(q|e)$  in model  $m$  is **polynomial** time in the **domain sizes** of the logical variables in  $q, e, m$

1.5 Attends(person)  $\rightarrow$  Series

1.2 Topic  $\rightarrow$  Attends(person)

# What is Lifted Inference?

- What is commonly understood as exact lifted inference?

## Definition: Domain-Lifted Inference

Complexity of computing  $P(q|e)$  in model  $m$  is **polynomial** time in the **domain sizes** of the logical variables in  $q, e, m$

- Possibly exponential in the size of  $q, e, m$ 
  - # predicates, # parfactors, # atoms,
  - # arguments, # formulas, # constants in model

# What is Lifted Inference?

- Motivation: Large domains lead to intractable propositional inference.
- A **formal framework** for lifted inference
  - Definition + complexity considerations
  - ~ PAC-learnability (Valiant)
- Other notions, e.g., for approximate inference.

# Completeness

- A procedure that is domain-lifted for all models in a class  $M$  is called **complete** for  $M$

*All models in  $M$  are “liftable”*

- There was **no completeness result** for existing algorithms

*If you give me a model,  
I cannot say if grounding will be needed,  
untill I run the inference algorithm itself.*



# Completeness Result

## Probabilistic inference in models with

- universal quantifiers  $\forall$  and
- 2 logical variables per formula

is domain-liftable.

- A non-trivial class of models
- **First completeness results** in exact lifted inference
  - Lifted knowledge compilation procedure
  - Lifted variable elimination procedure

# Completeness Game

Expressivity

No domain-lifted inference procedure exists

FOL  $\forall, \exists, =$  [Jaeger 99]

... [Jaeger 12]

?

FOL  $\forall, =$ , 2 variables [Van den Broeck 11]

Complete domain-lifted inference procedure

# Outline

- Introduction to lifted inference
- Four contributions
  - Arbitrary constraints
  - Completeness result
  - **Conditioning**
  - Approximate inference

# Conditioning

- Task: Probability of query  $q$  given evidence  $e$ :  $P(q|e)$   
Domain-lifted inference is exponential in the size of  $e$ .
- Can we compute **conditional probabilities** efficiently?  
Depends on the arity of literals conditioned on:

Literal Arity	Complexity of Conditioning
0	Polynomial
1	Polynomial if supported by compilation
$\geq 2$	#P-hard

- Positive and negative result for lifted inference

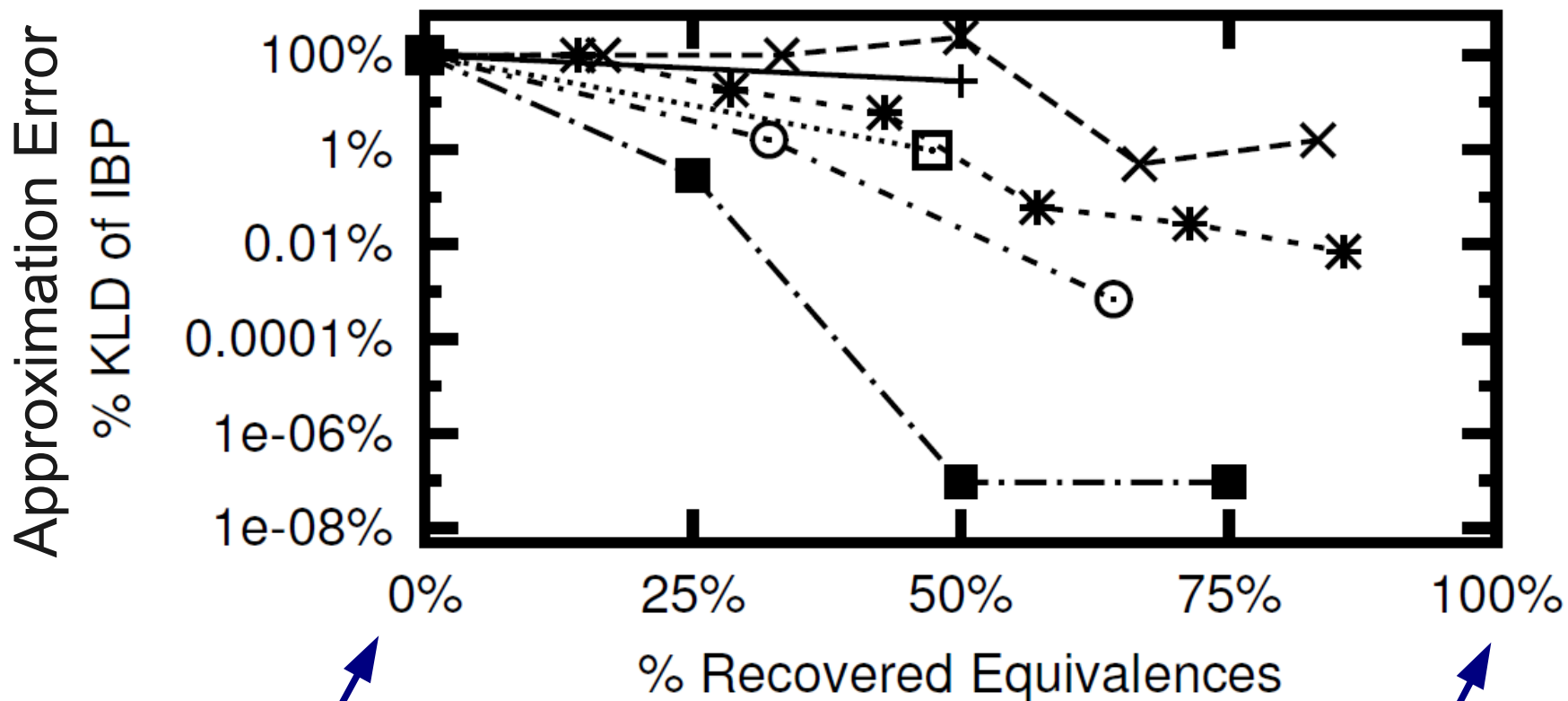
# Outline

- Introduction to lifted inference
- Four contributions
  - Arbitrary constraints
  - Completeness result
  - Conditioning
  - **Approximate inference**

# Lifted RCR

- **Practical** usefulness of lifted inference shown for **approximate** inference with lifted BP
- Lifted Relax, Compensate and Recover
  - (1) Clone all atoms in a model
  - (2) Relax equivalences between clones
  - (3) Compensate for removed equivalences
  - (4) Recover equivalences until model too complex
- Exact lifted inference black box in (3)

# Lifted RCR



Special case: Lifted BP  
Tractable

Exact lifted inference  
Intractable

# Outline

- Introduction to lifted inference
- Four contributions
  - Arbitrary constraints
  - Completeness result
  - Conditioning
  - Approximate inference



# Posters!

## Lifted Variable Elimination with Arbitrary Constraints

Nima Taghipour, Daan Fierens, Jesse Davis and Hendrik Blockeel

KATHOLIEKE UNIVERSITEIT  
**LEUVEN**

nima.taghipour@cs.kuleuven.be

### Context and Motivation

#### Probabilistic Logical Models

- Uncertainty: probability theory
- Relational structure: first-order logic

#### Efficient inference is a key challenge

Lifted inference: exploit the symmetries in such models

Identify a group of interchangeable variables, perform inference once for the group, instead of once for each individual

The groups are defined by means of constraints.

- Existing approaches use (inequality) constraints

### Contribution and Conclusion

We generalize lifted inference to work with arbitrary constraints

- capture more symmetries, i.e., more lifting

Empirically demonstrate that this

- improves efficiency with orders of magnitude
- allows for exact inference where only approximate inference was feasible

### Parfactor Models

Random variable (randvar): ground atom  $\rightarrow$   $\text{Smokes}(\text{ann})$

Parameterized randvar (PRV): a constrained atom

$\text{Smokes}(X) | X \in \{\text{ann}, \text{bob}, \text{carl}\}$

Logical variable  $D(X) = \{\text{ann}, \text{bob}, \dots, \text{zack}\}$

Constraint: an arbitrary subset of  $D(X)$

represents the group of randvars:

$\text{Smokes}(\text{ann}), \text{Smokes}(\text{bob}), \text{Smokes}(\text{carl})$

Parametric factor (parfactor): represents a group of factors

$\phi_i(S, A(X)) | X \in \{\text{ann}, \dots, \text{zack}\}$

$\phi_i(A(X), T) | X \in \{\text{ann}, \dots, \text{zack}\}$

Counting formula: exploit symmetry among interchangeable randvars

$A(\text{ann}), A(\text{bob}), A(\text{carl}), A(\text{dave})$

$\#X \in \{\text{ann}, \dots, \text{dave}\} [A(X)]$

### Lifted Variable Elimination

Variable elimination (VE): Eliminate randvar  $V$ :

- (1) multiply the factors involving  $V$  into one
- (2) sum-out  $V$  from the resulting factor

Lifted VE: do the same at the level of PRVs and parfactors

Algorithm: GC-FOVE - generalized C-FOVE [Mich et al. 2008]

A set of lifted operations:

- Sum-out
- Multiplication
- Counting Conversion
- Constraint handling operations (our main contribution)

One lifted operation:

many ground operations

$$\phi(S, A(X)) \sum_{S \in \mathcal{D}_S} \phi(S)$$

$$\phi(S, A(X)) \otimes \phi(A(X), T) \rightarrow \phi(S, T, A(X))$$

$$\phi(A(X)) \rightarrow \phi(\#X [A(X)])$$

$$\phi(S, A(X)) \sum_{S \in \mathcal{D}_S} \phi(S, A(X))$$

$$\phi(S, A(X)) \sum_{S \in \mathcal{D}_S} \phi(S, A(X))$$

$$\phi(S, A(X)) \sum_{S \in \mathcal{D}_S} \phi(S, A(X))$$

$$\phi(S, A(X)) \sum_{S \in \mathcal{D}_S} \phi(S, A(X))$$

$$\phi(S, A(X)) \sum_{S \in \mathcal{D}_S} \phi(S, A(X))$$

$$\phi(S, A(X)) \sum_{S \in \mathcal{D}_S} \phi(S, A(X))$$

$$\phi(S, A(X)) \sum_{S \in \mathcal{D}_S} \phi(S, A(X))$$

### Constraint Handling Operations

Rewrite the model so that it satisfies the preconditions of lifted operations

- Defined in terms of relational algebra operations

#### Splitting and Shattering

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

$\phi(A(X), A(Y)) | X \in \{\text{ann}, \text{bob}, \text{carl}\}, Y \in \{\text{ann}, \text{bob}, \text{carl}\}$

### Acknowledgements

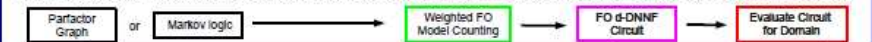
This research is supported by GOAD0008 'Probabilistic Logic Learning'. Daan Fierens is a postdoctoral fellow of the research foundation Flanders (FWO-Vlaanderen).

## First-Order Knowledge Compilation for Lifted Probabilistic Inference

[IJCAI11] Van den Broeck, Guy; Taghipour, Nima; Meert, Wannes; Davis, Jesse; De Raedt, Luc. Lifted probabilistic inference by first-order knowledge compilation [NIPS11] Van den Broeck, Guy. On the completeness of first-order knowledge compilation for lifted probabilistic inference [AAAI12] Van den Broeck, Guy; Davis, Jesse. Conditioning in first-order knowledge compilation and lifted probabilistic inference

### First-Order Knowledge Compilation [IJCAI11]

Compile a probabilistic logical model into a logical circuit and perform lifted probabilistic inference by evaluating the circuit.



Weight-Probability Formula in First-Order Logic

$2 \text{ friends}(X, Y) \wedge \text{smokes}(X) \Rightarrow \text{smokes}(Y)$

Logical Variable Domain of constants e.g.  $X$  in  $\{\text{alice}, \text{bob}\}$

Atom Random variable in  $\{\text{true}, \text{false}\}$  for each  $X$

Represents factor graph for given domain  $\{\text{alice}, \text{bob}\}$



factor  $\{ \text{friends}(\text{alice}, \text{bob}), \text{smokes}(\text{alice}), \text{smokes}(\text{bob}) \}$

$$P(q|e, M) = \frac{\text{WMC}(q \wedge e \wedge M)}{\text{WMC}(e \wedge M)}$$

Reduce probabilistic inference to weighted model counting in logic



set disjunction

decomposable conjunction

$\text{smokes}(X), X \in D$

$\neg \text{smokes}(X), X \notin D$

$\neg \text{friends}(X, Y), X \in D, Y \notin D$

Weighted model counting inference is efficient (polynomial) in a FO d-DNNF circuit.

### Completeness [NIPS11]

Factor graph explodes

e.g., 50 people: 2500 factors, 2650 random variables.

Propositional inference is intractable

Solution: Lifted inference

"Reason about groups of objects as a whole"

"Exploit symmetries" "Avoid repeated computations"

Research Questions:

• What is commonly understood as lifted inference?

• Contribution: A formal framework for exact lifted inference (definition + complexity considerations) – PAC learning

Domain-Lifted Probabilistic Inference

The complexity of computing  $P(q|e)$  in model  $M$  is

– Polynomial time in the domain sizes of the logical variables in  $q, e, M$

– Possibly exponential in the size of  $q, e, M$

# predicates, # partition, # atoms, # arguments, # formulas, # clauses, # constants in model

• When can a model be lifted?

"If you give me a model, I cannot say if grounding will be needed, until I run the inference algorithm itself."

• Contribution: A new operator for knowledge compilation

• Contribution: First completeness result ("liftability")

First-order knowledge compilation performs domain-lifted inference on any model with 2 logical variables per formula or parfactor (2-WFOMC).

Lifted inference is "solved" for 2-WFOMC, a first non-trivial class of problems.

### Conditioning [AAAI12]

Conditioning a logic formula  $\Sigma$  on term  $\gamma$  replaces the atoms of  $\gamma$  in  $\Sigma$  by true or false.

All propositional circuits can be conditioned in polytime.

Research Questions:

• Can we efficiently condition a first-order d-DNNF circuit?

• Contribution: An algorithm for conditioning on arity 0/1 relations

• Contribution: Conditioning on arity  $\geq 2$  is #P-hard, because any #2SAT problem can be solved by conditioning

• Contribution: First completeness result ("liftability")

Complexity of Conditioning

Polynomial

Polynomial if supported by compilation

#P-hard

• Can lifted inference compute conditional probabilities polynomially in the size of evidence?

In any first-order probabilistic model with a minimal expressivity, computing conditional probabilities exactly is #P-hard.

LEUVEN

FWO

FWO

<http://dtai.cs.kuleuven.be/ml/systems/wfomc>