Outline

- Part 1: Motivation
- Part 2: Probabilistic Databases
- Part 3: Weighted Model Counting
- Part 4: Lifted Inference for WFOMC

- Part 5: Completeness of Lifted Inference
- Part 6: Query Compilation
- Part 7: Symmetric Lifted Inference Complexity
- Part 8: Open-World Probabilistic Databases
- Part 9: Discussion & Conclusions

Question 2. Are lifted rules stronger than grounded?

Alternative to lifting:

1. Ground the FO sentence
2. Do WMC on the propositional formula

- There is no reason why grounded inference should be weaker than lifted inference
- However, *existing* grounded algorithms are strictly weaker than lifted inference

Algorithms for Model Counting

[Gomes'08] Based on full search DPLL:

- Shannon expansion.
  $\#F = \#F[X=0] + \#F[X=1]$
- Caching.
  Store $\#F$, look it up later
- Components. If $\mathrm{Vars}(F1) \cap \mathrm{Vars}(F2) = \emptyset$:
  $\#(F1 \wedge F2) = \#F1 * \#F2$

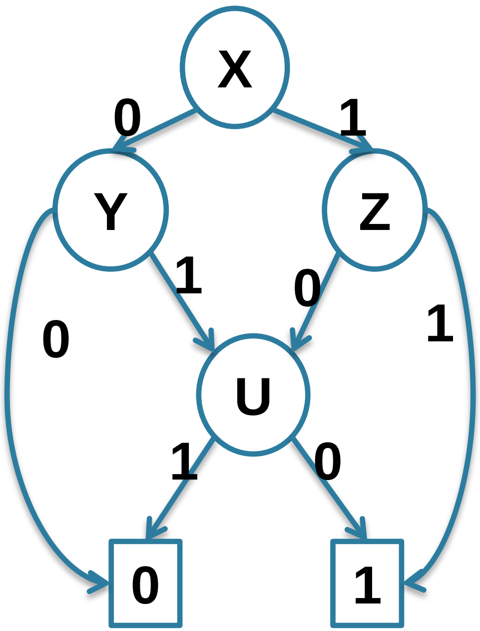
Knowledge Compilation

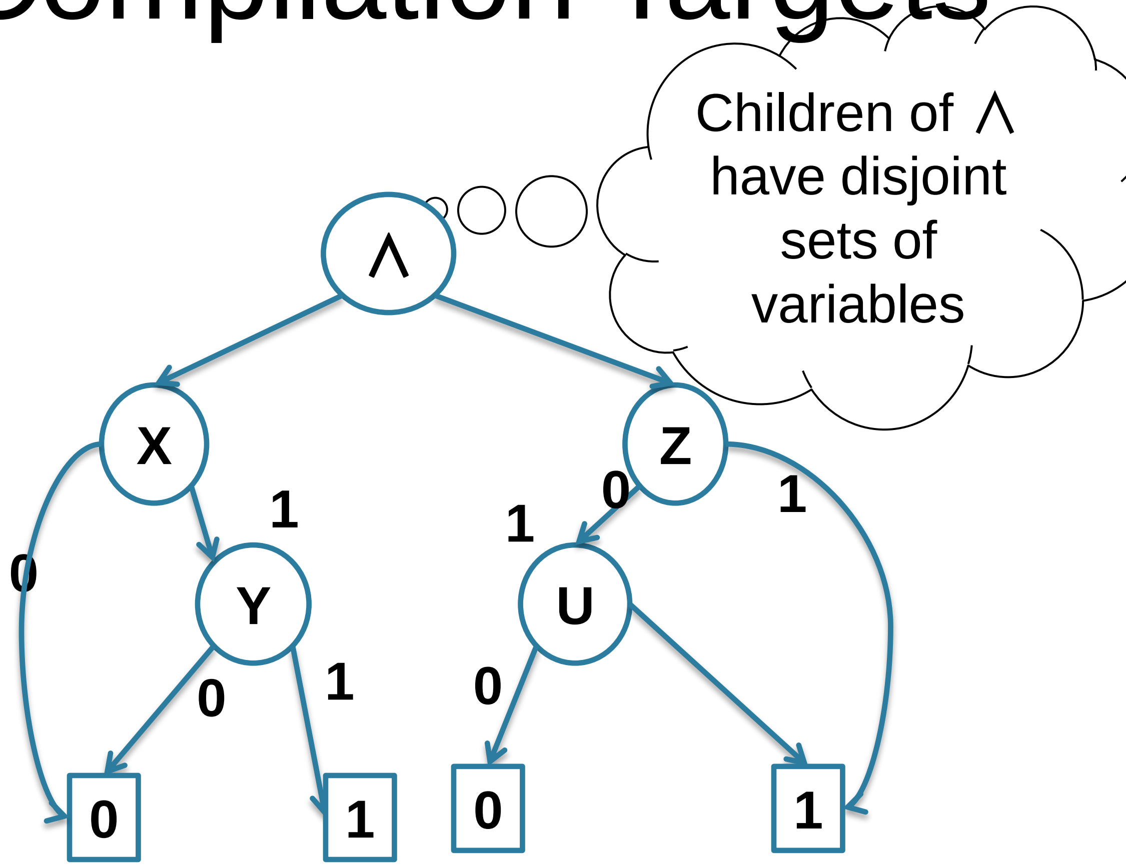
Definition (informal): represent the Boolean formula F in a circuit where WMC(F) is in PTIME in the size of the representation

Why we care:

- The trace of any inference algorithm is a knowledge compilation
- Lower bounds on size(KC) give lower bounds on the algorithm's runtime

Knowledge Compilation Targets





FBDD:
Decision-, sink-nodes

OBDD: fixed variable order

Decision-DNNF
add: ∧-nodes

[Huang&Darwiche'2005]

DPLL and Knowledge Compilation

Fact: Trace of full-search DPLL → KC:

- Basic DPLL
  - → decision trees
- DPLL + caching
  - → OBDD (fixed variable order)
  - → FBDD
- DPLL + caching + components
  - → decision-DNNF

Hard Queries

H_0 = $\forall x \forall y\ (R(x) \vee S(x,y) \vee T(y))$ = non-hierarchical
H_k = hierarchical, has inversion, for $k \geq 1$

Grounded Boolean formulas:
$F_n(H_0) = \wedge_{i \in [n],\, j \in [n]} (R_i \vee S_{ij} \vee T_j)$

Th. [Beame'14] Any FBDD for $F_n(H_k)$ has size $\geq 2^{n-1}/n$.
Same holds for any non-hierarchical query.

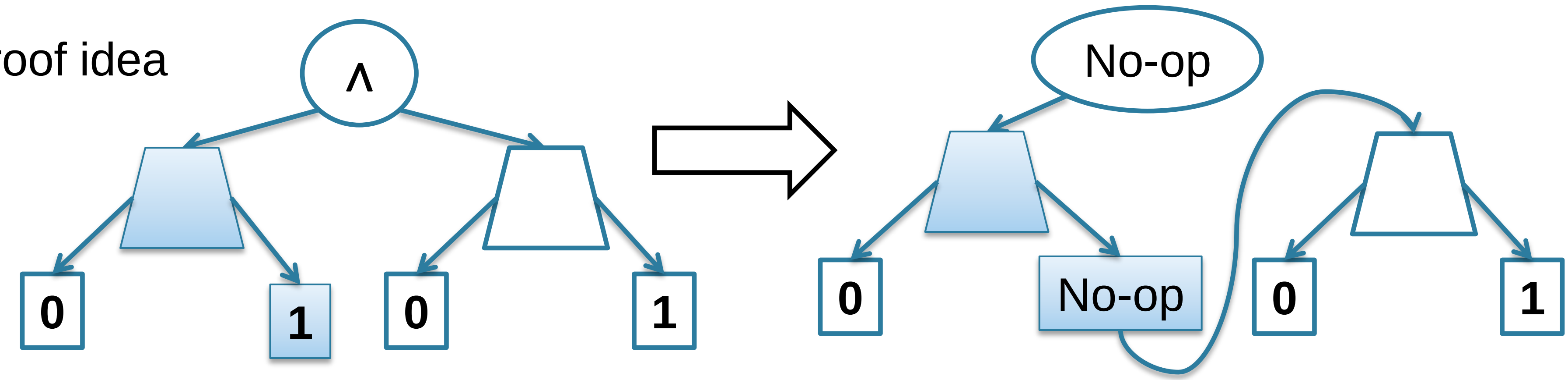
What about Decision-DNNFs?

Decision-DNNF to FBDD

Optimal [Razgon]

Theorem If F has a Decision-DNNF with N nodes, then F has an FBDD with at most $N^{1+\log(N)}$ nodes.

Proof idea

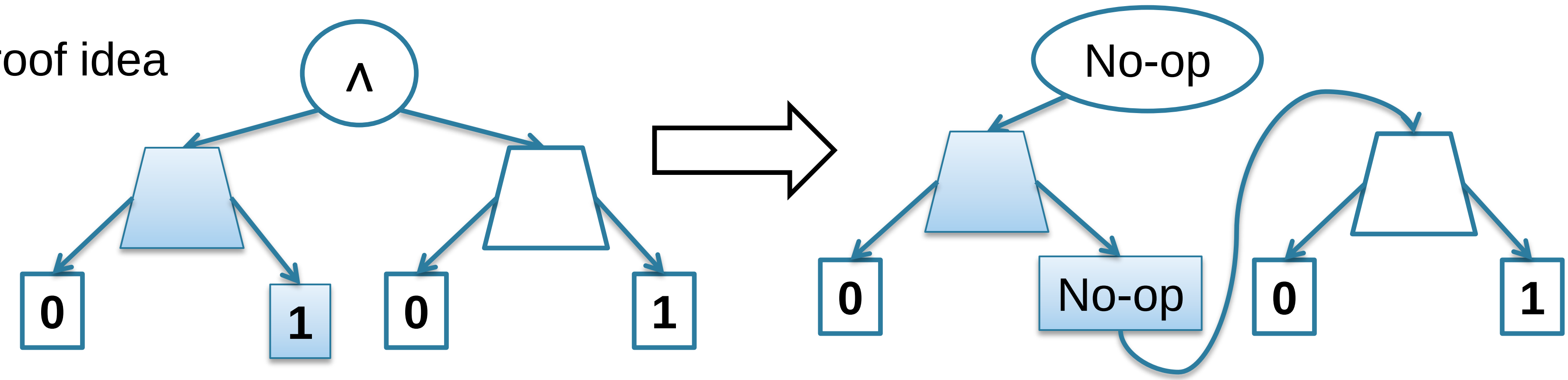


Decision-DNNF to FBDD

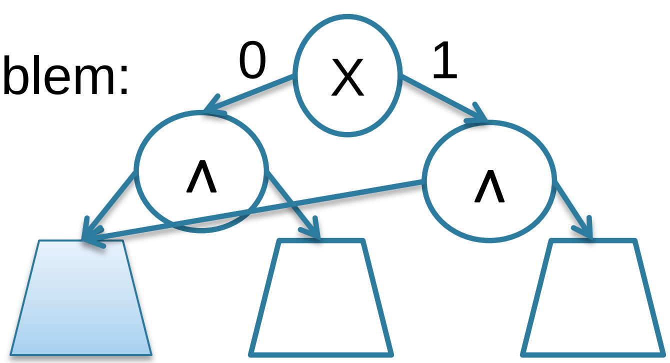
Optimal [Razgon]

Theorem If F has a Decision-DNNF with N nodes, then F has an FBDD with at most $N^{1+\log(N)}$ nodes.

Proof idea



Problem:

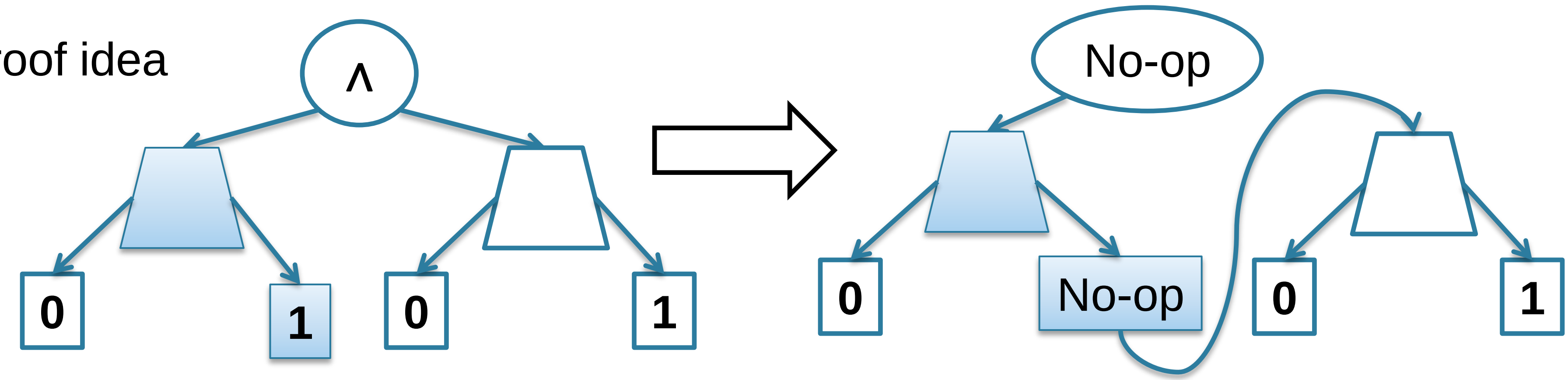


Decision-DNNF to FBDD

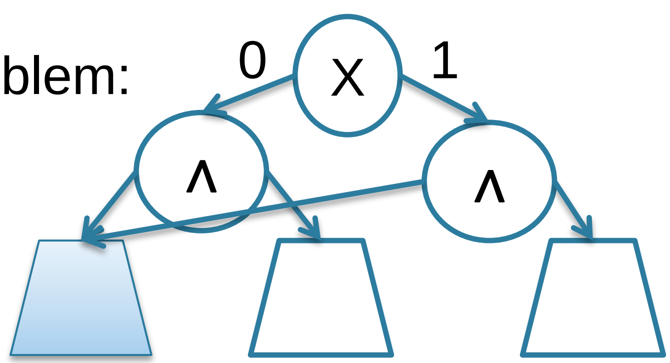
Optimal [Razgon]

Theorem If F has a Decision-DNNF with N nodes, then F has an FBDD with at most $N^{1+\log(N)}$ nodes.

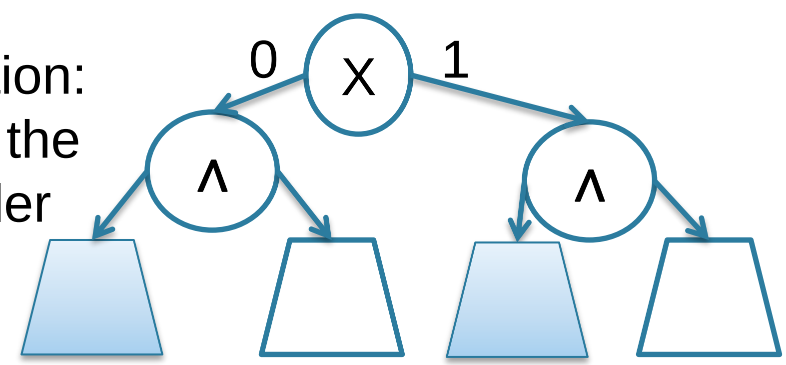
Proof idea



Problem:



Solution: copy the smaller child



Hard Queries

Corollary Any Decision-DNNF for $F_n(H_k)$ has size $2^{\Omega(\sqrt{n})}$
Same holds for any non-hierarchical query.

Proof. N-node Decision-DNNF to $N^{1+\log(N)}$ nodes FBDD.

$$N^{1+\log(N)} > 2^{n-1}/n \,,$$
$$\log(N) + \log^2(N) > n - 1 - \log(n)$$
$$\log^2(N) = \Omega(n)$$
$$\log(N) = \Omega(\sqrt{n})$$

Lifted v.s. Grounded Inference

	Non-hierarchical Q (e.g. H_0)
Lifted P(Q)	#P-hard
Grounded $P(F_n(Q))$	$2^{\Omega(\sqrt{n})}$

What about hierarchical queries ?

Inversion-Free Queries

Definition An inversion in Q is a sequence of co-occurring vars:

$(x_0,y_0), (x_1,y_1), \ldots, (x_k,y_k),$ such that:

- $at(x_0) \not\subseteq at(y_0)$, $at(x_1)=at(y_1), \ldots, at(x_{k-1})=at(y_{k-1})$, $at(x_k) \not\supseteq at(y_k)$
- For all i=1,..,k-1 there exists two atoms in Q of the form:
  $S_i(\ldots,x_{i-1},\ldots,y_{i-1},\ldots)$ and $S_i(\ldots,x_i, \ldots, y_i, \ldots)$

Inversion-free implies hierarchical, but converse fails

$Q=[R(x_0) \vee S(x_0,y_0)] \wedge [S(x_1,y_1) \vee T(x_1)]$

Inversion-free

Inversion

$H_1=[R(x_0) \vee S(x_0,y_0)] \wedge [S(x_1,y_1) \vee T(y_1)]$

Easy Queries

[Jha&S.11], [Beame'15]

Theorem Let Q in $\forall FO^{un}$

1. If Q has inversion then OBDD for $F_n(Q)$ has size $\geq 2^{n-1}/n$
2. Else, $F_n(Q)$ has OBDD of width $2^{\#atoms(Q)}$ (size $O(n)$)

Proof (part 2 only – next slide)

Easy Queries

[Beame&Liew'15] Extended to SDD.
Thus, over $\forall FO^{un}$, OBDD ≈ SDD

[Jha&S.11], [Beame'15]

Theorem Let Q in $\forall FO^{un}$

1. If Q has inversion then OBDD for $F_n(Q)$ has size $\geq 2^{n-1}/n$
2. Else, $F_n(Q)$ has OBDD of width $2^{\#atoms(Q)}$ (size $O(n)$)

Proof (part 2 only – next slide)

Easy Queries

[Bova'16] SDD more succint than OBDD (HWB)

[Beame&Liew'15] Extended to SDD.
Thus, over $\forall FO^{un}$, OBDD ≈ SDD

[Jha&S.11], [Beame'15]

Theorem Let Q in $\forall FO^{un}$

1. If Q has inversion then OBDD for $F_n(Q)$ has size $\geq 2^{n-1}/n$
2. Else, $F_n(Q)$ has OBDD of width $2^{\#atoms(Q)}$ (size $O(n)$)

Proof (part 2 only – next slide)

$Q = [R(x) \vee S(x,y)] \wedge [T(x') \vee S(x',y')]$

$$Q = [R(x) \vee S(x,y)] \wedge [T(x') \vee S(x',y')]$$

$n = 2$

$$\Pi = \underbrace{R_1 T_1 S_{11} S_{12}}_{x = 1} \underbrace{R_2 T_2 S_{21} S_{22}}_{x = 2}$$

$\boxed{C_1 = R(x) \vee S(x,y)} \wedge \boxed{C_2 = T(x') \wedge S(x',y')} = \boxed{Q = [R(x) \vee S(x,y)] \wedge [T(x') \vee S(x',y')]}$

$n = 2$

$\Pi = \underbrace{R_1 T_1 S_{11} S_{12}}_{x = 1} \underbrace{R_2 T_2 S_{21} S_{22}}_{x = 2}$

$\boxed{C_1 = R(x) \vee S(x,y)} \wedge \boxed{C_2 = T(x') \wedge S(x',y')} = \boxed{Q = [R(x) \vee S(x,y)] \wedge [T(x') \vee S(x',y')]}$

$F_2(C_1) = (R_1 \vee S_{11}) \wedge (R_1 \vee S_{12}) \wedge (R_2 \vee S_{21}) \wedge (R_2 \vee S_{22})$

$n = 2$

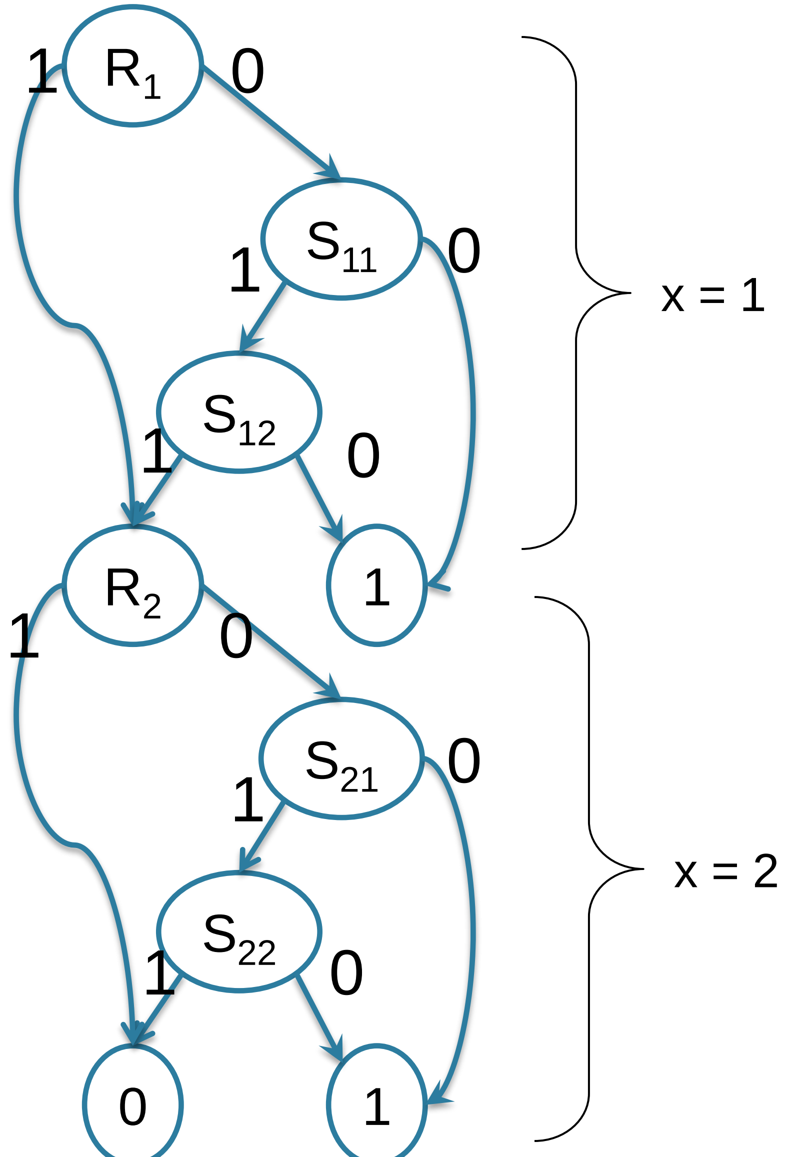
$\Pi = \underbrace{R_1 T_1 S_{11} S_{12}}_{x = 1} \underbrace{R_2 T_2 S_{21} S_{22}}_{x = 2}$

$\boxed{C_1 = R(x) \vee S(x,y)} \wedge \boxed{C_2 = T(x') \ \wedge S(x',y')} = \boxed{Q = [R(x) \vee S(x,y)] \wedge [T(x') \vee S(x',y')]}$

$F_2(C_1) = (R_1 \vee S_{11}) \wedge (R_1 \vee S_{12}) \wedge (R_2 \vee S_{21}) \wedge (R_2 \vee S_{22})$

$n = 2$

$\Pi = \underbrace{R_1 T_1 S_{11} S_{12}}_{x=1} \underbrace{R_2 T_2 S_{21} S_{22}}_{x=2}$

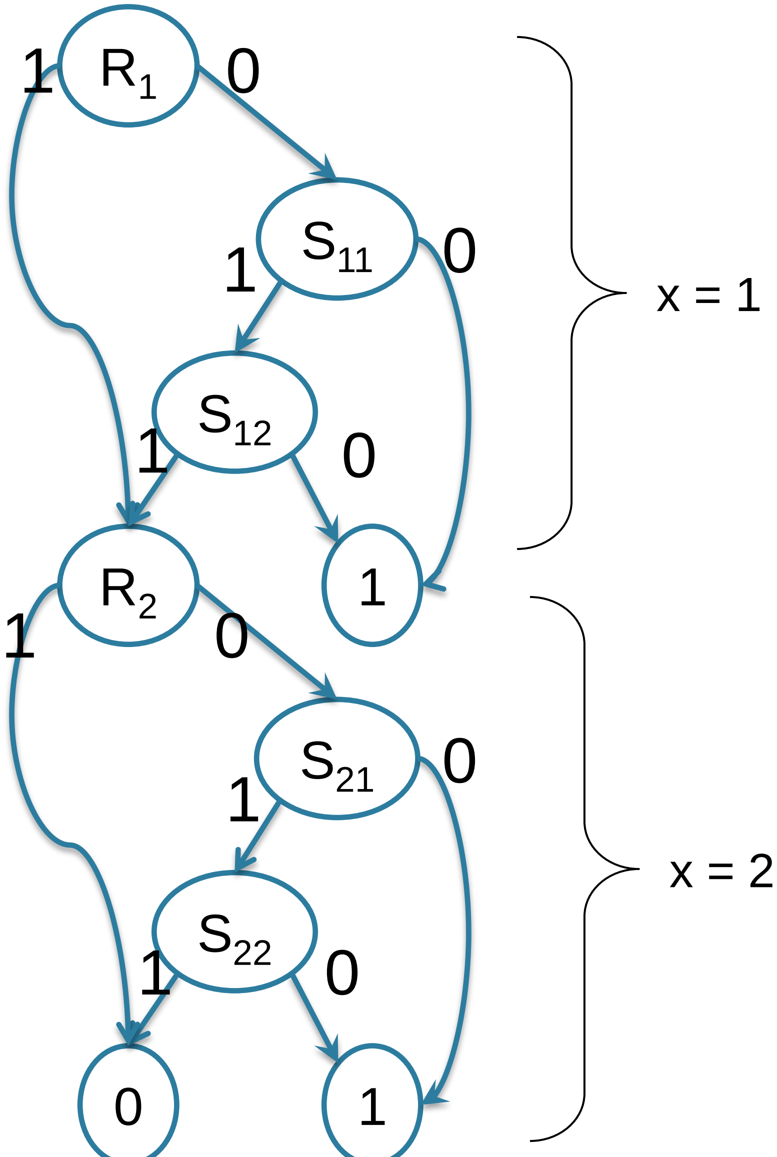


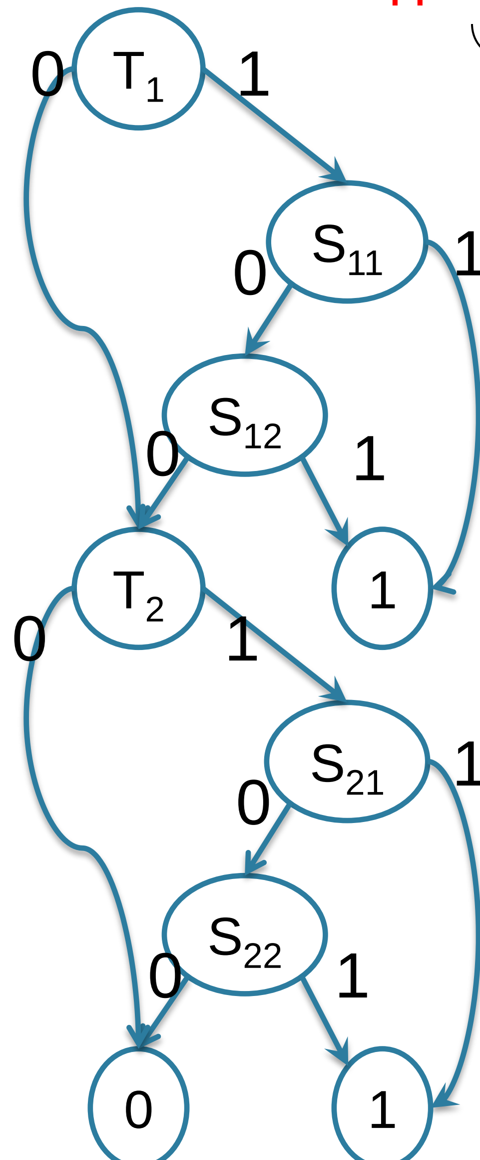
$\boxed{C_1 = R(x) \vee S(x,y)} \wedge \boxed{C_2 = T(x') \wedge S(x',y')} = \boxed{Q = [R(x) \vee S(x,y)] \wedge [T(x') \vee S(x',y')]}$

$F_2(C_1) = (R_1 \vee S_{11}) \wedge (R_1 \vee S_{12}) \wedge (R_2 \vee S_{21}) \wedge (R_2 \vee S_{22})$

$n = 2$

$\Pi = \underbrace{R_1 T_1 S_{11} S_{12}}_{x = 1} \underbrace{R_2 T_2 S_{21} S_{22}}_{x = 2}$





Same variable order Π in both OBDDs!

OBDD for
$Q = C_1 \wedge C_2$
has width =
width1 × width2

Lifted v.s. Grounded Inference

	Non-hierarchical Q (e.g. H_0)	Inversion-free Q
Lifted P(Q)	#P-hard	PTIME
Grounded $P(F_n(Q))$	$2^{\Omega(\sqrt{n})}$	PTIME

Easy/Hard Queries

Main result: a class of queries Q such that:

- Lifted inference: P((Q)) in PTIME
- Grounded inference: $P(F_n(Q))$ exponential time

Significance: limitation of DPLL-based algorithms for model counting

Clauses of H_k

$H_{k0} = \forall x \forall y\ R(x) \vee S_1(x,y)$

$H_{k1} = \forall x \forall y\ S_1(x,y) \vee S_2(x,y)$

$H_{k2} = \forall x \forall y\ S_2(x,y) \vee S_3(x,y)$

…

…

$H_{kk} = \forall x \forall y\ S_k(x,y) \vee T(y)$

Clauses of H_k

$H_{k0} = \forall x \forall y\ R(x) \vee S_1(x,y)$
$H_{k1} = \forall x \forall y\ S_1(x,y) \vee S_2(x,y)$
$H_{k2} = \forall x \forall y\ S_2(x,y) \vee S_3(x,y)$
…
…
$H_{kk} = \forall x \forall y\ S_k(x,y) \vee T(y)$

$f(Z_0, Z_1, \ldots, Z_k)$ = a Boolean function

Clauses of H_k

$H_{k0} = \forall x \forall y\, R(x) \vee S_1(x,y)$
$H_{k1} = \forall x \forall y\, S_1(x,y) \vee S_2(x,y)$
$H_{k2} = \forall x \forall y\, S_2(x,y) \vee S_3(x,y)$
…
…
$H_{kk} = \forall x \forall y\, S_k(x,y) \vee T(y)$

$f(Z_0, Z_1, \ldots, Z_k)$ = a Boolean function

$Q = f(H_{k0}, H_{k1}, \ldots, H_{kk})$

Clauses of H_k

$H_{k0} = \forall x \forall y\ R(x) \vee S_1(x,y)$
$H_{k1} = \forall x \forall y\ S_1(x,y) \vee S_2(x,y)$
$H_{k2} = \forall x \forall y\ S_2(x,y) \vee S_3(x,y)$
…
…
$H_{kk} = \forall x \forall y\ S_k(x,y) \vee T(y)$

$f(Z_0, Z_1, \ldots, Z_k)$ = a Boolean function

$Q = f(H_{k0}, H_{k1}, \ldots, H_{kk})$

Examples:

$f = Z_0 \wedge Z_1 \wedge \ldots \wedge Z_k$ then $f(H_{k0}, H_{k1}, \ldots, H_{kk}) = H_k$

$f = Z_0 \wedge Z_2 \vee Z_0 \wedge Z_3 \vee Z_1 \wedge Z_3$ then $f(H_{30}, H_{31}, H_{31}, H_{33}) = Q_W$

Easy/Hard Queries

[Beame'14]

Theorem For any Boolean function $f(Z_0, Z_1, \ldots, Z_k)$, denoting $Q = f(H_{k0}, H_{k1}, \ldots, H_{kk})$:

- Any FBDD for $F_n(Q)$ has size $2^{\Omega(n)}$
- Any Decision-DNNF has size $2^{\Omega(\sqrt{n})}$.

Consequence:

- Lifted inference computes compute $P(Q_W)$ in PTIME
- Any DPLL-based algorithm takes time $2^{\Omega(\sqrt{n})}$

Many other queries are like Q_W

Lifted v.s. Grounded Inference

	Non-hierarchical Q (e.g. H_0)	Inversion-free Q	Q = $f(H_{k0},\ldots,H_{kk})$
Lifted P(Q)	#P-hard	PTIME	PTIME or #P-hard
Grounded $P(F_n(Q))$	$2^{\Omega(\sqrt{n})}$	PTIME	$2^{\Omega(\sqrt{n})}$

Two Questions

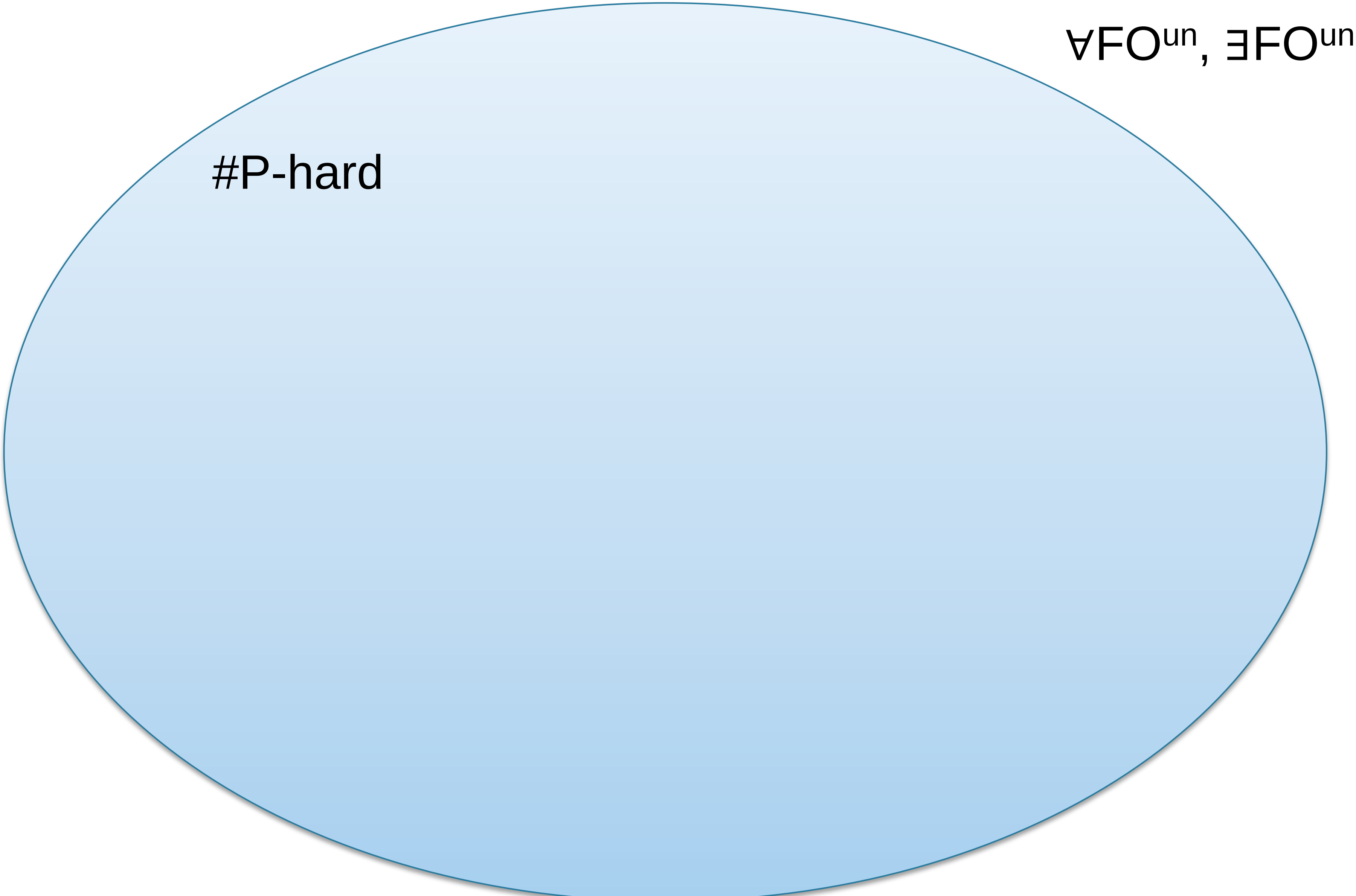
- Question 1: Are the lifted rules complete?
  - We know that they get stuck on some queries
  - Should we add more rules?

Complete for "unate ∀FO" and for "unate ∃FO"

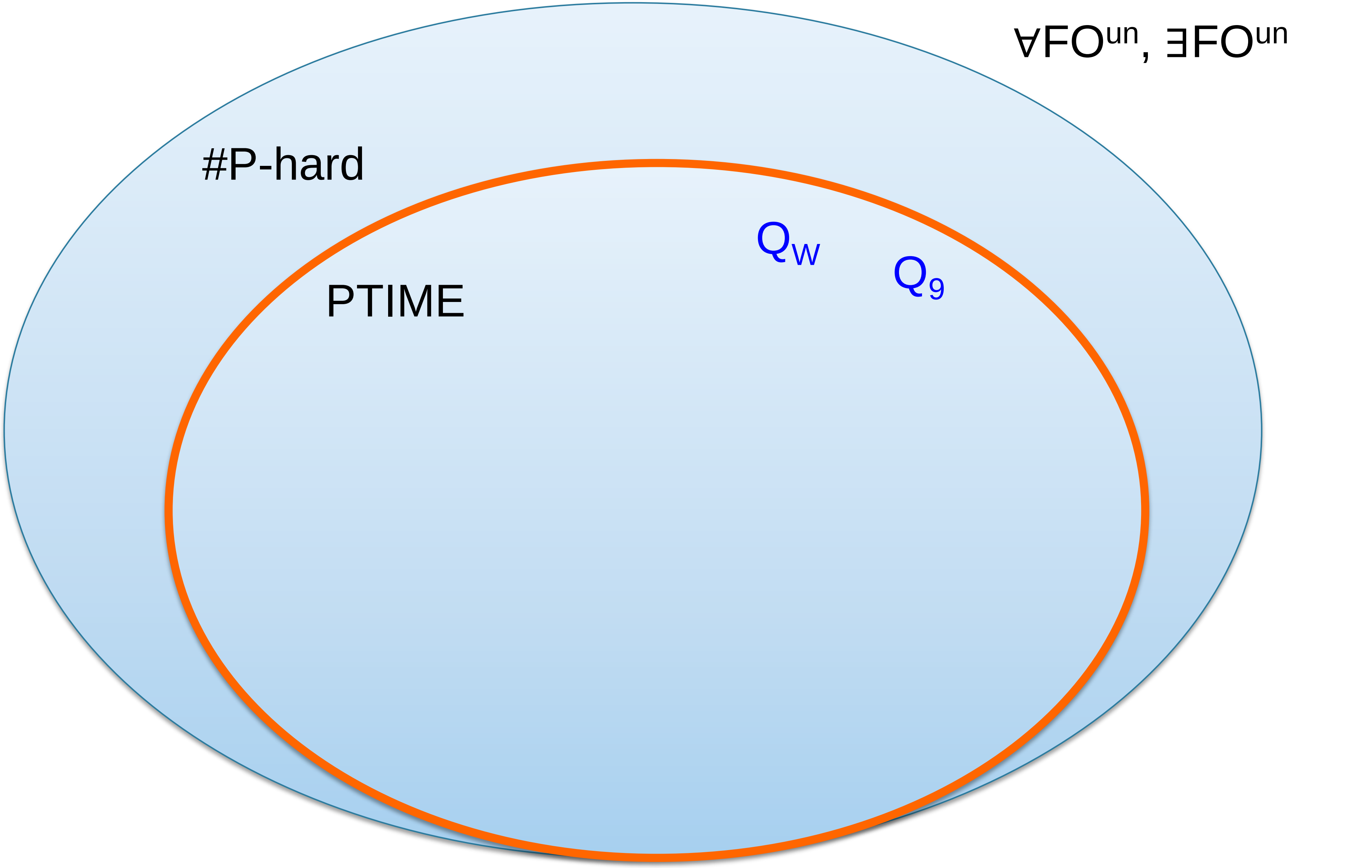
- Question 2: Are lifted rules stronger than grounded?
  - Lifted rules can also be grounded
  - Any advantage over grounded inference?

Strictly stronger than DPLL-based algorithms

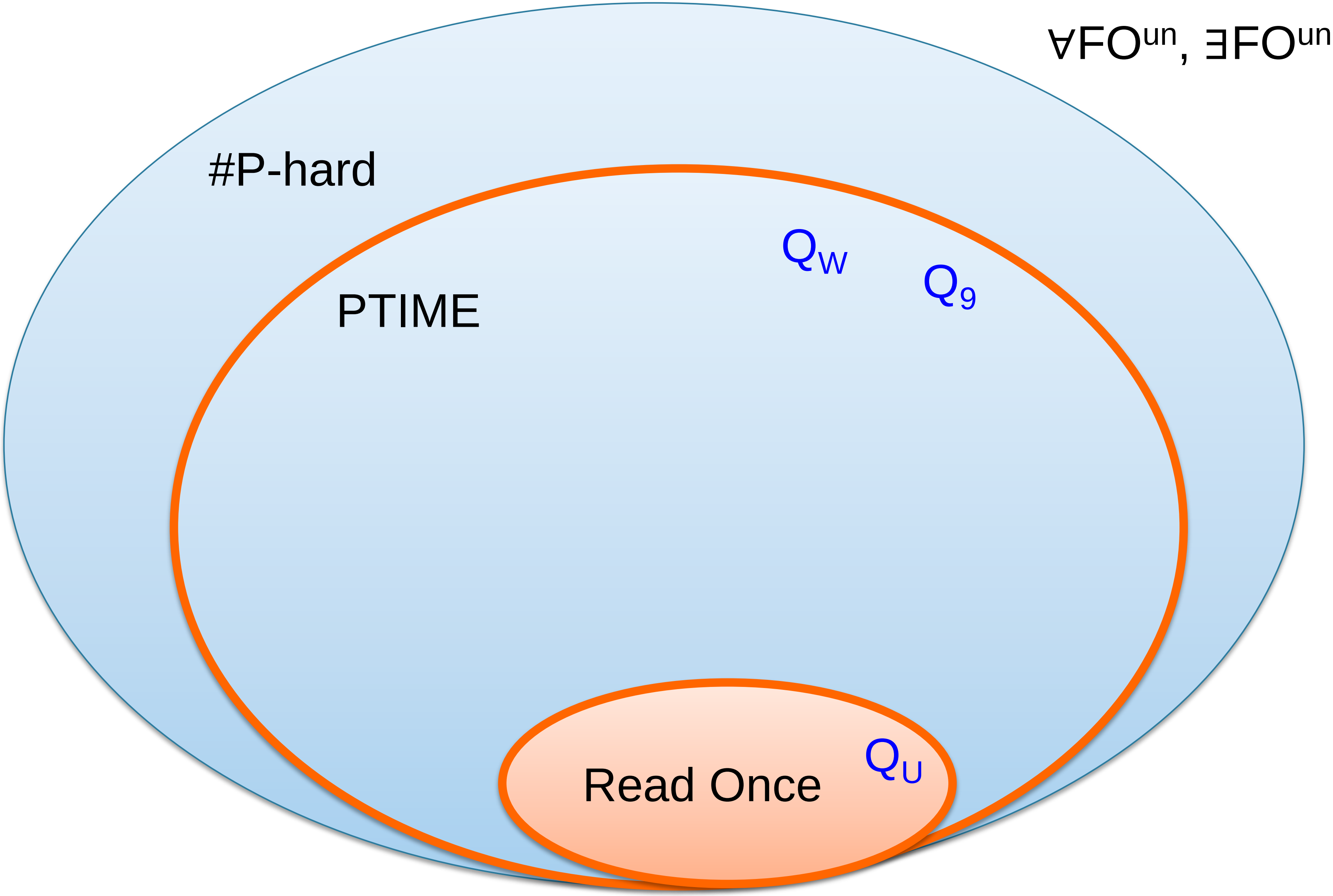
Möbius Über Alles



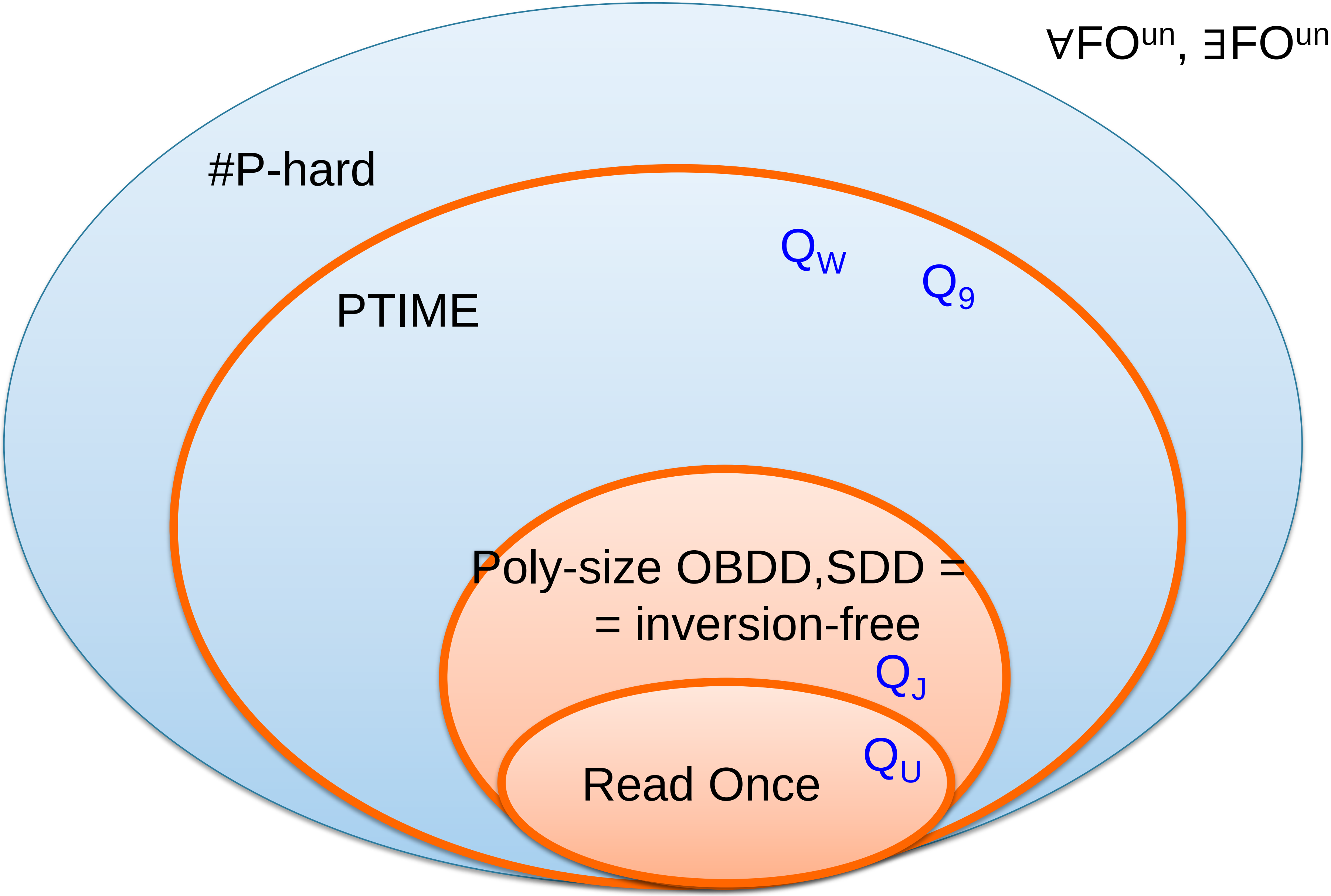
Möbius Über Alles



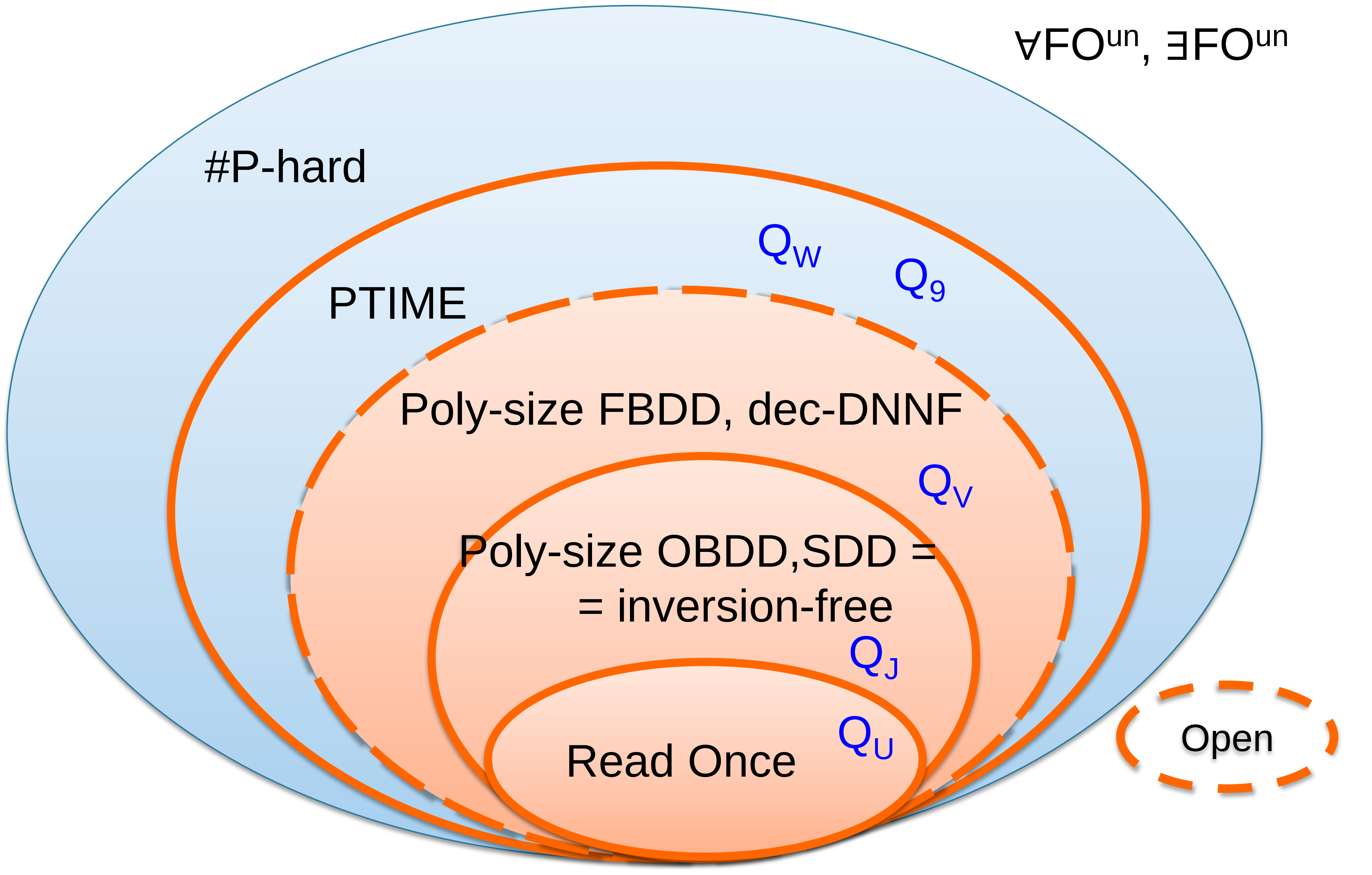
Möbius Über Alles



Möbius Über Alles



Möbius Über Alles



Möbius Über Alles