Approximate Counting CSPs

Andrei A. Bulatov
Simon Fraser University

Fields Institute 2011

Constraint Satisfaction

Let Γ be a set of relations

#CSP(Γ):

Instance: (V,$\mathcal{C}$).

Objective: How many solutions does (V,$\mathcal{C}$) have?

Let $\mathcal{B}$ be a relational structure

#CSP(B):

Instance: A relational structure $\mathcal{A}$ of the same type as $\mathcal{B}$.

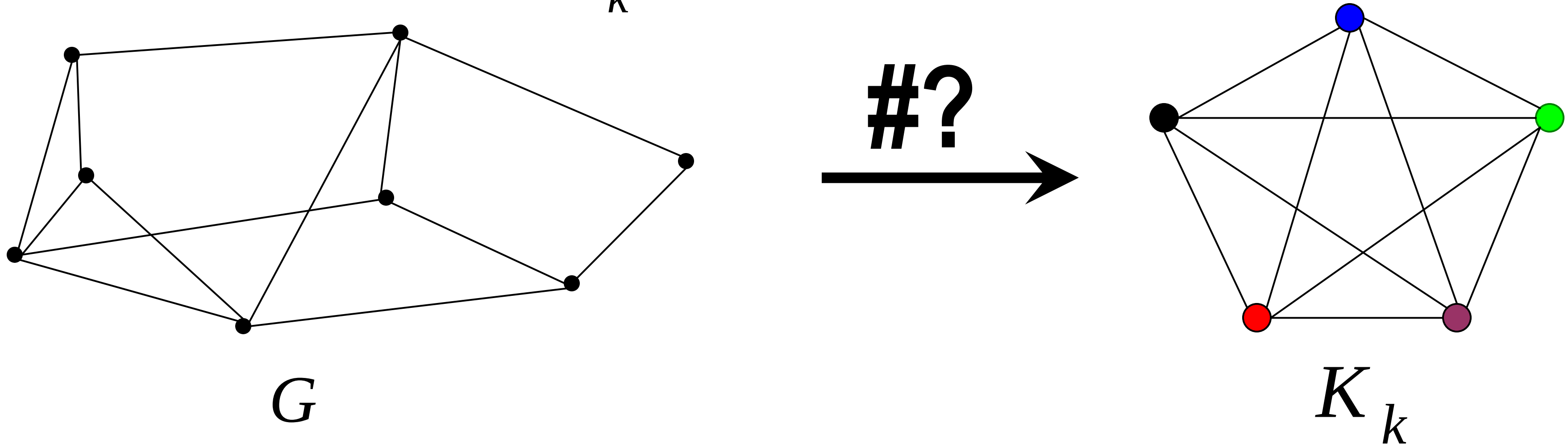
Objective: How many homomorphisms from $\mathcal{A}$ to $\mathcal{B}$ are there?

Counting Homomorphisms

#k-Coloring:

Instance: A graph G.

Objective: How many homomorphisms from G to K_k are there?



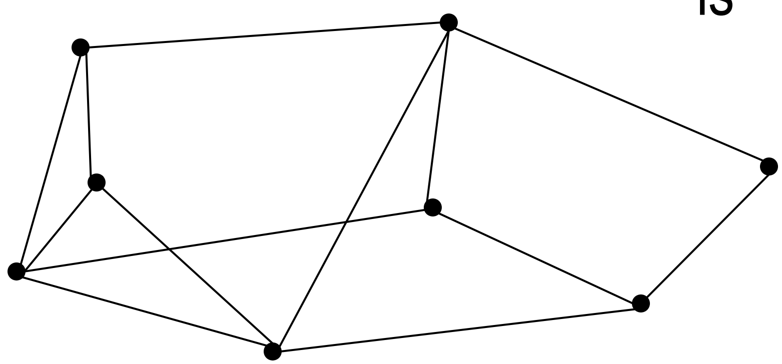
More general: Let H be a (di)graph **#H-Coloring = #CSP(H)**

Counting Homomorphisms

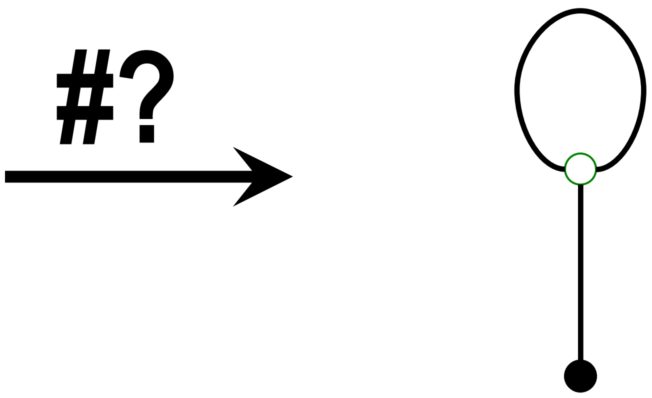
#Independent Set:

Instance: A graph G.

Objective: How many homomorphisms from G to H_{is} are there?



G



H_{is}

Examples: #SAT, Linear Equations

#3-SAT: = **#CSP(** $\mathcal{C}_3$ **)**

Instance: A propositional formula $\Phi = C_1 \wedge \ldots \wedge C_n$ in 3-CNF.

Objective: How many satisfying assignments are there?

#Linear Equations: = **#CSP(** $\mathcal{F}$ **)**

Instance: A system of linear equations

$$\begin{cases} a_{11}x_1 + \ldots + a_{1m}x_m = b_1 \\ \ldots \\ a_{n1}x_1 + \ldots + a_{nm}x_m = b_n \end{cases}$$

Objective: How many solutions are there?

Counting Problems: Fourier Coefficients

Let $f:\{0,1\}^n \rightarrow \{0,1\}$ be a Boolean operation and

$$S = \{i_1,\ldots,i_k\} \subseteq \{1,\ldots,n\}$$

Fourier coefficient $\hat{f}(S)$ is given by

$$\hat{f}(S) = \frac{1}{2^n} \sum_{x_1,\ldots,x_n \in \{0,1\}^n} (-1)^{f(x_1,\ldots,x_n)} (-1)^{x_{i_1}+\cdots+x_{i_k}}$$

Observe that computing $\hat{f}(S)$ reduces to counting the zeroes of

$$f(x_1,\ldots,x_n) + x_{i_1} + \cdots + x_{i_k}$$

Weighted #CSP

Γ is a set of functions f: $B \to \mathbb{R}$ (natural, real, complex)
Instance of #CSP(Γ): (V,$\mathcal{C}$), $\mathcal{C}$ is a set of `constraints' $f(\vec{x})$

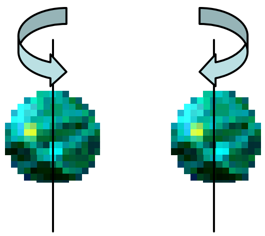
Given an instance I = (V,$\mathcal{C}$) the weight of a mapping
σ: $V \to B$ is computed as

$$w(\sigma) = \prod_{f(\vec{u}) \in \mathcal{C}} f(\sigma(\vec{u}))$$

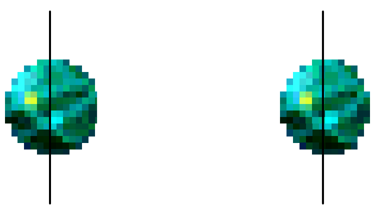
Then

$$Z(I) = \sum_{\sigma: V \to B} w(\sigma)$$

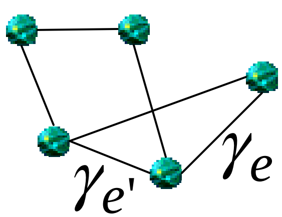
Spin Systems



A particle can have one of the two spins

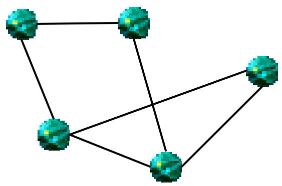


Two particles interact iff they have the same spin



A system of interacting particles form a graph with edge weights $\gamma_e > -1$, a spin system

Potts Model



A configuration of the spin system S is an assignment of spins $\sigma: S \to \{L,R\}$

Energy of the system is $\frac{1}{H(\sigma)} = \prod_{e=(u,v)\text{ an edge}} (1+\gamma_e(\sigma(u)=\sigma(v))$

The probability the system is in configuration σ equals

$\frac{1}{Z}\left(\frac{1}{T \cdot H(\sigma)}\right)$ where $Z = \sum_\sigma \frac{1}{T \cdot H(\sigma)}$ is the partition function, and T is temperature (Gibbs distribution)

$$Z_{\text{Potts}} = \sum_\sigma \prod_{e=(u,v)\text{ an edge}} (1+\gamma_e(\sigma(u)=\sigma(v))$$

Potts Model (cntd)

Thus Potts model is equivalent to #CSP(Γ) where Γ contains all binary functions of the form

$$\begin{pmatrix} 1+\gamma & 1 \\ 1 & 1+\gamma \end{pmatrix}, \quad \gamma \in \mathbb{R}$$

Exact Counting: CSP

Theorem (B.; 2008)
For any Γ the problem #CSP(Γ) is either polynomial time solvable, or #P-complete

Theorem (Dyer, Richerby.; 2010)
Given Γ, the problem of deciding if #CSP(Γ) is poly time or not is in NP.

Exact Counting: Weighted CSP

Corollary (B.; 2008)
For any Γ of non-negative rational valued functions the problem #CSP(Γ) is either polynomial time solvable, or #P-complete

Theorem (Cai, Chen; 2010)
For any Γ of non-negative real valued functions the problem #CSP(Γ) is either poly time solvable, or #P-complete

Theorem (B.,Dyer,Goldberg, et al.; 2009)
For any Γ of functions $f: \{0,1\}^k \to \mathbb{Q}$ the problem #CSP(Γ) is either polynomial time solvable, or #P-complete

Approximation

Relative error:

$$\Pr[e^{-\varepsilon} Z(I) \leq A(I) \leq e^{\varepsilon} Z(I)] \geq 3/4$$

An FPRAS: given I and ε, output $A(I)$ satisfying the inequality above in time polynomial in $|I|$ and ε^{-1}

Sampling

Given an instance of CSP and a probability distribution on its solutions, output a solution according to the distribution

- output a random 3-coloring
- output an independent set with probability proportional to its size

An approximate sampler: given I, distribution π_I and ε, outputs σ according to probability distribution ω_I such that the statistical distance between ω_I and π_I is less than ε, an runs in time polynomial in $|I|$ and ε^{-1}

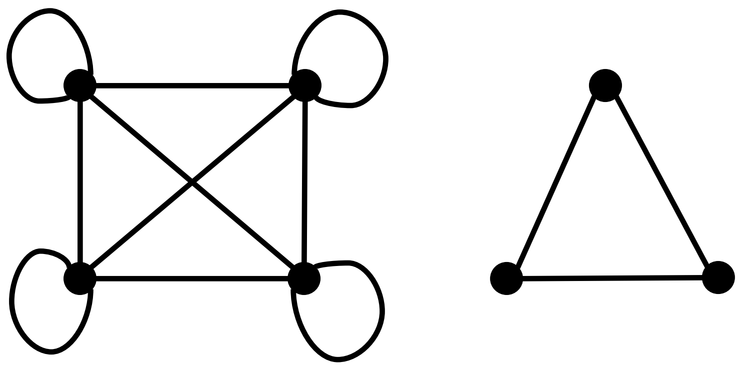
Counting and Sampling

Theorem (Jerrum,Valiant,Vazirani; 1986)
If #CSP(Γ) is self-reducible then an FPRAS exists iff an approximate sampler exists

Let I be an instance. CSP is self reducible if we can fix value of a variable.
If we can count, choose a variable v, and let N_d be the number of solutions with $v = d$. Choose value of v according to distribution $N_d, d \in D$, substitute, and recurse.
If we can sample, estimate the probability $p_{v=d} = N_d / N$.
Substitute any value with nonzero probability, and recurse.
The answer is $p_{v_1=d_1} \cdot p_{v_2=d_2} \cdot \dots \cdot p_{v_n=d_n}$

Easy Problems

Problems that can be solved exactly are easy
There are other easy problems



Other easy problems from Potts model, solved using Markov chains (see Jerrum et al.)
Also #Match and #DNF

AP-Reduction

An AP-reduction from P to P' is a randomized algorithm A solving P using an oracle for P'.

Input: (I,ε), I an instance of P

Oracle call: (J,δ), $\delta^{-1} \leq \mathrm{poly}(|I|, \varepsilon^{-1})$

Running time: polynomial in $|I|$ and ε^{-1}

Requirements: If the oracle is an FPRAS, A must be an FPRAS

Hard Problems

Hardest: #SAT AP-interreducible

#SAT has no FPRAS unless NP = RP (Zuckermann, 1996)

Theorem (Dyer et al; 2003)
#SAT is #P-complete with respect to AP-reductions

AP-interreducible with #SAT:

#IS (independent sets)

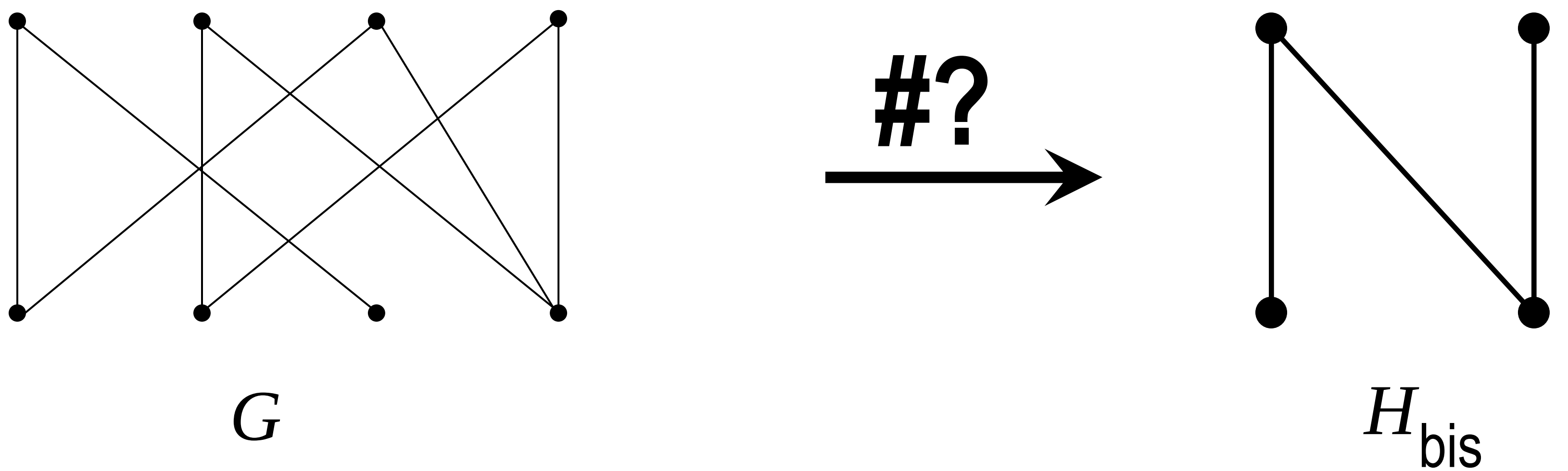
#MaxIS

Widom-Rowlingson configurations

#BIS

#Bipartite-Independent-Set (#BIS): Given a bipartite graph, find the number of its independent sets

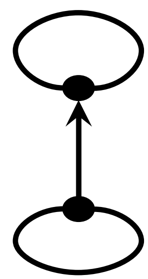
= #CSP(H_{bis})



#BIS and Friends

#BIS is not believed to have FPRAS or be #SAT AP-interreducible
Many other problems are interreducible with #BIS
#Downset: Given a poset, find the number of downsets in it

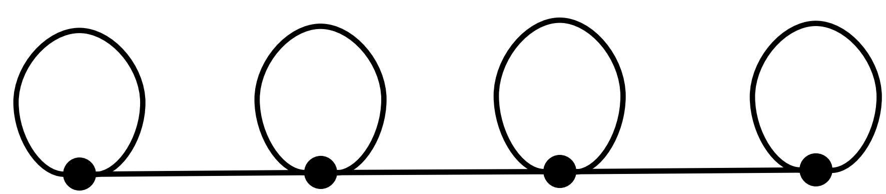
#CSP(H_{ds})



H_{ds}

#1p1nSAT Given a CNF such that every clause has a positive and a negative literal, find the number of satisfying assignments
#BeachConfigs

#CSP(H_{bc}) H_{bc}



More #BIS

Theorem (B., Hedayaty, 2010)
Let $\mathcal{A}$ be a relational structure. Then if it has both meet and join operations of a distributive lattice as polymorphisms then **#CSP($\mathcal{A}$)** is AP-reducible to #BIS.

Datalog

A Datalog program is a finite set of rules of the form

head body

$$T(\bar{x}) :- S_1(\bar{y}_1), \ldots, S_r(\bar{y}_r)$$

relational symbols

Let $H = (V,E)$ be a graph

$$\begin{aligned} T(x,y) &:- \quad E(x,y) \\ T(x,y) &:- \quad E(x,z), T(z,y) \end{aligned}$$

A Datalog program is linear if each rule contains at most one auxiliary predicate in the body

Datalog

A fixed point of a Datalog program is a value of $T(x,y)$ such that all the rules are satisfied

#FixedPoints(P): Given an instance I of Datalog program P, find the number of fixed points of P on I

Theorem (Dyer et al.; 2003)
A problem is reducible to #BIS if and only if it is AP-interreducible with #FixedPoints(P) for some Datalog program P.

Boolean Approximation

Theorem (Dyer,Goldberg,Jerrum, 2007)

Let $\mathcal{A}$ be a relational structure over {0,1}. Then

- if $\mathcal{A}$ has a Mal'tsev polymorphism, then **#CSP($\mathcal{A}$)** is solvable in polynomial time;
- otherwise, if it has both conjunction and disjunction as polymorphisms then **#CSP($\mathcal{A}$)** is as hard as #BIS;
- otherwise it is hard.

Between FPRAS and #BIS 1

Theorem (Bordewich; 2010)
If FPRAS $<_{AP}$ #BIS $<_{AP}$ #P then there is an infinite hierarchy of classes not AP-reducible to each other.

Theorem
If H is a reflexive oriented graph then **#CSP(H)** is #BIS-hard.

Between FPRAS and #BIS 2

Theorem (Goldberg,Kelk,Paterson; 2004)
If there is an approximate sampler for H-colorings (H is a `nontrivial' undirected graph), then there is a sampler for BIS.

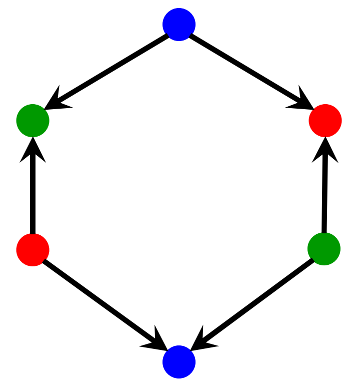
Since BIS is self-reducible

Corollary
For an undirected graph H, #CSP(H) is either in FPRAS or is #BIS-hard.

Beyond Trichotomy

#Bipartite 3-Colorability = $\#\mathrm{CSP}\begin{pmatrix} a & a & b & b & c & c \\ 1 & 2 & 0 & 2 & 0 & 1 \end{pmatrix}$

Not believed to be #BIS-easy
or #P-hard



FPRAS $\leq_{AP}$ #BIS $\leq_{AP}$ #B3-COL $\leq_{AP}$ #B5-COL $\leq_{AP}$ ... $\leq_{AP}$ #SAT

Counting to Optimization

Observation

VCSP(Γ) $\leq_{AP}$ #CSP(Γ)

Take I an instance of VCSP(Γ)

Let I^k be the instance obtained by repeating all functions k times.

Then $Z(I^k) = \sum_{W \text{ a possible weight of a solution}} n_W W^k$

Choose k such that the maximal W dominates, $k = \text{poly}(I)$

Output $\dfrac{Z(I^{k+1})}{Z(I^k)}$

Classifications for Weighted #CSP

For any set Γ of functions $f: D^n \to \mathbb{R}$ we want to determine the complexity of #CSP(Γ)

Computable reals: There is a TM that given n computes the fist n bits of a in time poly(n)

Theorem (Yamakami; 2010)
Let Γ be a set of functions $f:\{0,1\}^n \to \mathbb{C}$ that contains all unary functions. Then #CSP(Γ) is either FPRAS, or #BIS, or #SAT-hard.

Reductions and Constructions

For a Γ, what functions f can be added to Γ so that

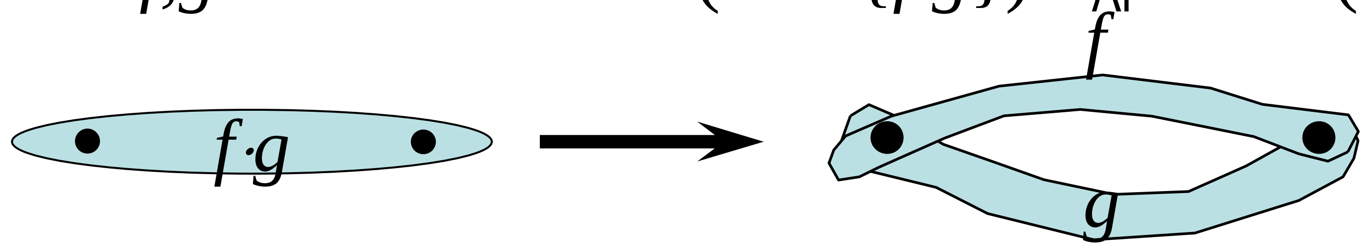
$$\#CSP(\Gamma \cup \{f\}) \leq_{AP} \#CSP(\Gamma) \quad ?$$

- Multiplying by a constant function:

  $f \in \Gamma$ then $\#CSP(\Gamma \cup \{\alpha \cdot f\}) \leq_{AP} \#CSP(\Gamma)$

Reductions and Constructions 2

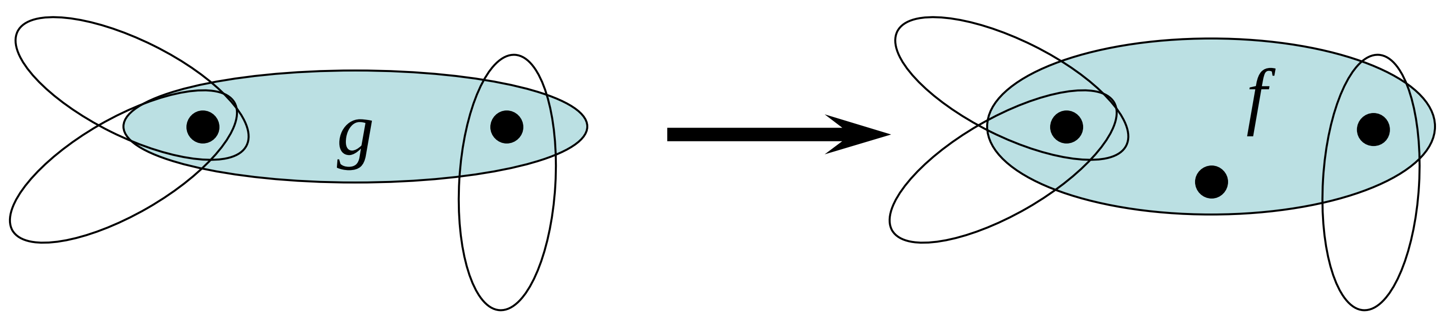
- Product: $f,g \in \Gamma$ then #CSP($\Gamma \cup \{f \cdot g\}$) $\leq_{AP}$ #CSP(Γ)



- Summation: $f(x_1,\ldots,x_n,y) \in \Gamma$ and

$$g(x_1,\ldots,x_n) = \sum_{y \in D} f(x_1,\ldots,x_n,y)$$

then #CSP($\Gamma \cup$ {g}) $\leq_{AP}$ #CSP(Γ)

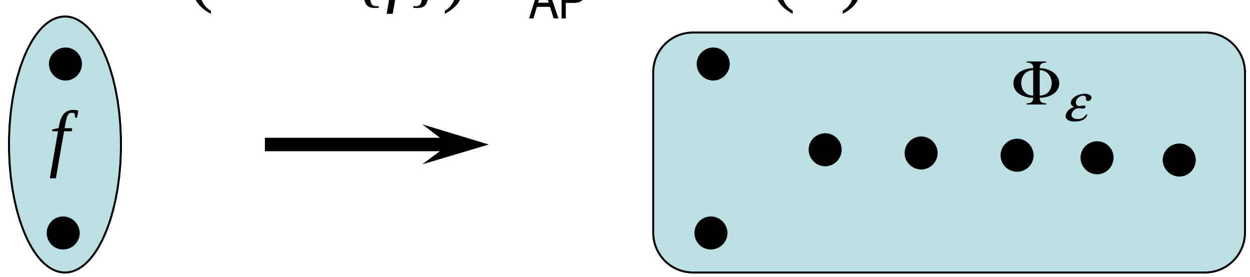


Reductions and Constructions 3

- Denote by $[\Gamma]$ the set of all functions obtained from Γ using the above operations;

  Call a sequence of such operations a pps-formula
- Limits: A function f is a limit of functions from Γ if
  - f is computable
  - for any $\varepsilon > 0$ there is f_ε such that $\| f - f_\varepsilon \|_\infty < \varepsilon$
  - there is a TM, poly time in ε^{-1} that computes pps-formulas for f_ε

  Then #CSP$(\Gamma \cup \{f\}) \le_{AP}$ #CSP(Γ)



Reductions and Constructions 4

A set of functions closed under multiplication by a constant, products, summation, and limits is said to be an ω-clone

The ω-clone generated by a set of functions Γ is denoted $\langle\Gamma\rangle$

Theorem (B.,Dyer,Goldberg,Jerrum; 2011)
If $\Gamma' \subseteq \langle\Gamma\rangle$ is finite then #CSP(Γ') $\leq_{AP}$ #CSP(Γ)

Log Supermodular Functions

A function $f:\{0,1\}^n \to \mathbb{R}$ is said to be log supermodular if for any $\vec{x},\vec{y} \in \{0,1\}^n$

$$f(\vec{x})\cdot f(\vec{y}) \le f(\vec{x}\wedge\vec{y})\cdot f(\vec{x}\vee\vec{y})$$

Lemma
LSM is an ω-clone

For any `nontrivial' function $f \in$ LSM, $H_{ds} \in \langle f \rangle$
Question 1: Does H_{ds} (+ some unary functions maybe) generate LSM?
Question 2: Any `morphisms' for ω-clones?

Log Supermodular Functions

Theorem (B.,Dyer,Goldberg,Jerrum; 2011)
Let Γ be a set of functions $f:\{0,1\}^n \rightarrow \mathbb{R}$ containing all nonnegative unary functions. Then either Γ is LSM or #CSP(Γ) is #P-complete

Corollary
Let Γ be a set of functions as above. Then either #CSP(Γ) is in FPRAS, or it is #BIS-hard or it is #P-complete

Thank you!