Arc Consistency

Manuel Bodirsky

CNRS / LIX, Ecole Polytechnique

December 2012

Local Consistency Techniques

Central idea: ‘constraint propagation’

> Perform local inferences;
> iterate

Local Consistency Techniques

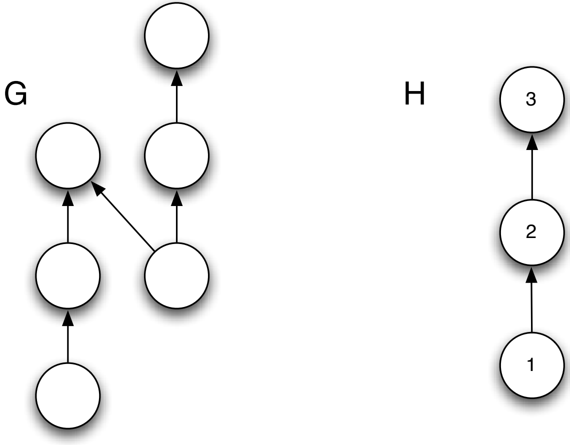
Central idea: ‘constraint propagation’

> Perform local inferences;
> iterate

- One of the earliest algorithmic approaches to the CSP
  Montanari’74, Mackworth’77, Freuder’82
- Very important in the AI literature
- Comes close to what humans naively try when confronted with a CSP

Example

Does G homomorphically map to $H := \vec{P}_2$?



The Arc-Consistency Procedure (AC)

Fix finite digraph H.

$\text{AC}_H(G)$
Input: a finite digraph G.
Data structure: a list $L(x) \subseteq V(H)$ for each vertex $x \in V(G)$.

Set $L(x) := V(H)$ for all $x \in V(G)$.
Do
    For each $(x, y) \in E(G)$:
        Remove u from $L(x)$ if there is no $v \in L(y)$ with $(u, v) \in E(H)$.
        Remove v from $L(y)$ if there is no $u \in L(x)$ with $(u, v) \in E(H)$.
        If $L(x)$ is empty for some vertex $x \in V(G)$ then **reject**
Loop until no list changes

The Arc-Consistency Procedure (AC)

Fix finite digraph H.

$\text{AC}_H(G)$
Input: a finite digraph G.
Data structure: a list $L(x) \subseteq V(H)$ for each vertex $x \in V(G)$.

Set $L(x) := V(H)$ for all $x \in V(G)$.
Do
  For each $(x, y) \in E(G)$:
    Remove u from $L(x)$ if there is no $v \in L(y)$ with $(u, v) \in E(H)$.
    Remove v from $L(y)$ if there is no $u \in L(x)$ with $(u, v) \in E(H)$.
    If $L(x)$ is empty for some vertex $x \in V(G)$ then **reject**
Loop until no list changes

Claim: if algorithm derives empty list, there is no homomorphism from A to B!

AC: nice features

- Implementations with linear running time (for fixed H)

AC: nice features

- Implementations with linear running time (for fixed H)
- Linear memory space (for fixed H)

AC: nice features

- Implementations with linear running time (for fixed H)
- Linear memory space (for fixed H)
- Easy to implement!

AC: nice features

- Implementations with linear running time (for fixed H)
- Linear memory space (for fixed H)
- Easy to implement!
- Does not only work for digraphs, but generalizes to relational structures ('hyperarc-consistency'; see script)

AC: nice features

- Implementations with linear running time (for fixed H)
- Linear memory space (for fixed H)
- Easy to implement!
- Does not only work for digraphs, but generalizes to relational structures ('hyperarc-consistency'; see script)
- Always a one-sided test

AC: nice features

- Implementations with linear running time (for fixed H)
- Linear memory space (for fixed H)
- Easy to implement!
- Does not only work for digraphs, but generalizes to relational structures ('hyperarc-consistency'; see script)
- Always a one-sided test

Problem (Incompleteness)

It might be that AC_H does not derive empty list on G, even though there is no homomorphism from G to H.

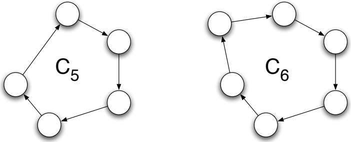
AC: nice features

- Implementations with linear running time (for fixed H)
- Linear memory space (for fixed H)
- Easy to implement!
- Does not only work for digraphs, but generalizes to relational structures ('hyperarc-consistency'; see script)
- Always a one-sided test

Problem (Incompleteness)

It might be that AC_H does not derive empty list on G, even though there is no homomorphism from G to H.

Example:



Completeness

Definition

Say that AC solves $\text{CSP}(H)$ if for all G:
AC_H derives empty list if and only if there is no homomorphism from G to H.

Completeness

Definition

Say that AC solves CSP(H) if for all G:
AC_H derives empty list if and only if there is no homomorphism from G to H.

- AC solves CSP($\vec{P}_2$), CSP($\vec{P}_k$)

Completeness

Definition

Say that AC solves CSP(H) if for all G:
AC_H derives empty list if and only if there is no homomorphism from G to H.

- AC solves CSP$(\vec{P}_2)$, CSP$(\vec{P}_k)$
- AC solves CSP(T_3), CSP(T_k)

Completeness

Definition

Say that AC solves CSP(H) if for all G:
AC_H derives empty list if and only if there is no homomorphism from G to H.

- AC solves CSP$(\vec{P}_2)$, CSP$(\vec{P}_k)$
- AC solves CSP(T_3), CSP(T_k)
- AC does not solve CSP(K_3) (no surprize...)

Completeness

Definition

Say that AC solves CSP(H) if for all G:
AC_H derives empty list if and only if there is no homomorphism from G to H.

- AC solves CSP($\vec{P}_2$), CSP($\vec{P}_k$)
- AC solves CSP(T_3), CSP(T_k)
- AC does not solve CSP(K_3) (no surprize...)
- AC does not solve CSP(C_k) (as we have seen)

Completeness

Definition

Say that AC solves CSP(H) if for all G:
AC_H derives empty list if and only if there is no homomorphism from G to H.

- AC solves CSP$(\vec{P}_2)$, CSP$(\vec{P}_k)$
- AC solves CSP(T_3), CSP(T_k)
- AC does not solve CSP(K_3) (no surprize...)
- AC does not solve CSP(C_k) (as we have seen)

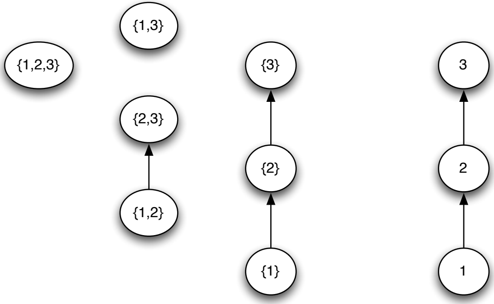
Question: for which H does AC solve CSP(H)?

The Power Graph

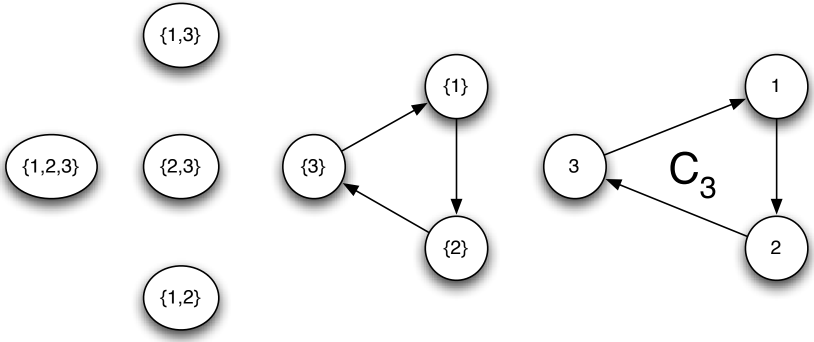
Definition (Power-Set Graph)

The power-set graph $P(H)$ of a digraph H has non-empty subsets of $V(H)$ as vertices. Put an edge between S_1 and S_2 iff

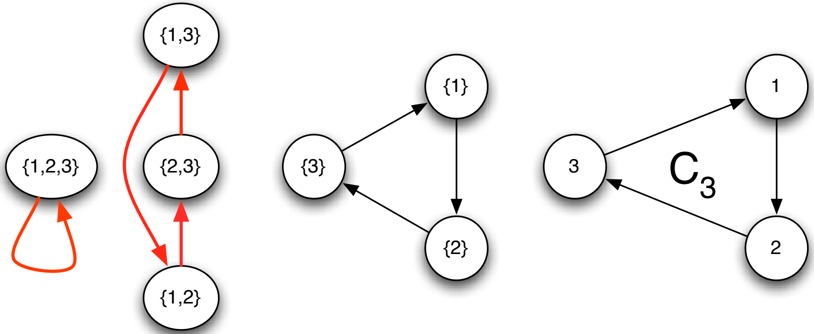
- for every $u \in S_1$ there exists a $v \in S_2$ such that $(u, v) \in E(H)$, and
- for every $v \in S_2$ there exists $u \in S_1$ such that $(u, v) \in E(H)$.



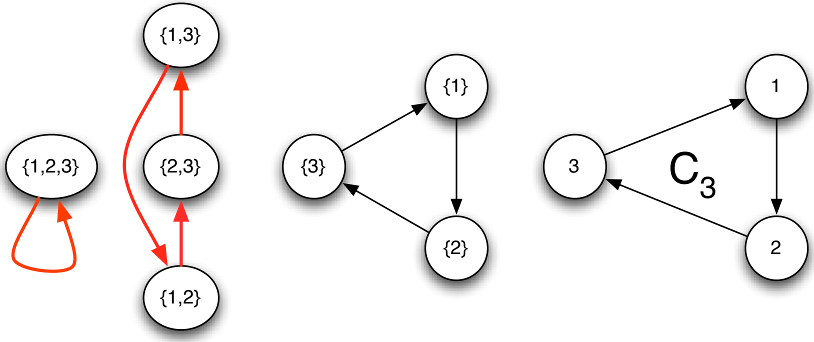
Another Example and Observation



Another Example and Observation

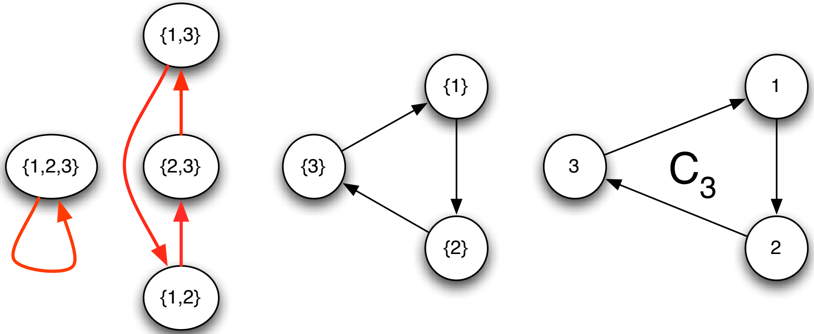


Another Example and Observation



- For all H we have $H \to P(H)$

Another Example and Observation



- For all H we have $H \to P(H)$
- Not for all H we have $P(H) \to H$.

Characterization

Theorem 1.

Let H be a finite digraph. The following are equivalent.

1. AC solves CSP(H)
2. $P(H)$ homomorphically maps to H

Examples:

Characterization

Theorem 1.

Let H be a finite digraph. The following are equivalent.

1. AC solves CSP(H)
2. $P(H)$ homomorphically maps to H

Examples:

- $\vec{P}(P_2) \rightarrow \vec{P}_2$

Characterization

Theorem 1.

Let H be a finite digraph. The following are equivalent.

1. AC solves CSP(H)
2. $P(H)$ homomorphically maps to H

Examples:

- $\vec{P}(P_2) \rightarrow \vec{P}_2$
- $P(T_3) \rightarrow T_3$

Characterization

Theorem 1.

Let H be a finite digraph. The following are equivalent.

1. AC solves CSP(H)
2. $P(H)$ homomorphically maps to H

Examples:

- $\vec{P}(P_2) \rightarrow \vec{P}_2$
- $P(T_3) \rightarrow T_3$
- $P(\vec{C}_3) \not\rightarrow \vec{C}_3$

Characterization

Theorem 1.

Let H be a finite digraph. The following are equivalent.

1. AC solves CSP(H)
2. $P(H)$ homomorphically maps to H

Examples:

- $\vec{P}(P_2) \to \vec{P}_2$
- $P(T_3) \to T_3$
- $P(\vec{C}_3) \not\to \vec{C}_3$

Lemma

AC_H does not derive the empty list on G if and only if $G \to P(H)$.

Proof of Lemma

Want to show: AC_H does not derive the empty list on G iff $G \to P(H)$.

Proof of Lemma

Want to show: AC_H does not derive the empty list on G iff $G \to P(H)$.

'$\Rightarrow$'

Let $L(x)$ be the (non-empty) set at the final stage of AC_H.

Claim: $x \mapsto L(x)$ is a homomorphism from G to $P(H)$.

Proof of Lemma

Want to show: AC_H does not derive the empty list on G iff $G \to P(H)$.

'$\Rightarrow$'

Let $L(x)$ be the (non-empty) set at the final stage of AC_H.
Claim: $x \mapsto L(x)$ is a homomorphism from G to $P(H)$.
Suppose $(a, b) \in E(G)$.

Proof of Lemma

Want to show: AC_H does not derive the empty list on G iff $G \to P(H)$.

'$\Rightarrow$'

Let $L(x)$ be the (non-empty) set at the final stage of AC_H.

Claim: $x \mapsto L(x)$ is a homomorphism from G to $P(H)$.

Suppose $(a, b) \in E(G)$. By Definition of AC_H:

Proof of Lemma

Want to show: AC_H does not derive the empty list on G iff $G \to P(H)$.

'$\Rightarrow$'

Let $L(x)$ be the (non-empty) set at the final stage of AC_H.

Claim: $x \mapsto L(x)$ is a homomorphism from G to $P(H)$.

Suppose $(a, b) \in E(G)$. By Definition of AC_H:

for each $u \in L(a)$ there is $v \in L(b)$ such that $(u, v) \in E(H)$.

for each $v \in L(b)$ there is $u \in L(a)$ such that $(u, v) \in E(H)$.

Proof of Lemma

Want to show: AC_H does not derive the empty list on G iff $G \to P(H)$.

'$\Rightarrow$'

Let $L(x)$ be the (non-empty) set at the final stage of AC_H.

Claim: $x \mapsto L(x)$ is a homomorphism from G to $P(H)$.

Suppose $(a, b) \in E(G)$. By Definition of AC_H:

for each $u \in L(a)$ there is $v \in L(b)$ such that $(u, v) \in E(H)$.

for each $v \in L(b)$ there is $u \in L(a)$ such that $(u, v) \in E(H)$.

By definition of $P(H)$, we have $(L(a), L(b)) \in E(P(H))$.

Proof of Lemma

Want to show: AC_H does not derive the empty list on G iff $G \to P(H)$.

'$\Rightarrow$'
Let $L(x)$ be the (non-empty) set at the final stage of AC_H.
Claim: $x \mapsto L(x)$ is a homomorphism from G to $P(H)$.
Suppose $(a, b) \in E(G)$. By Definition of AC_H:
for each $u \in L(a)$ there is $v \in L(b)$ such that $(u, v) \in E(H)$.
for each $v \in L(b)$ there is $u \in L(a)$ such that $(u, v) \in E(H)$.
By definition of $P(H)$, we have $(L(a), L(b)) \in E(P(H))$.

'$\Leftarrow$'
Let $f \colon G \to P(H)$ a homomorphism.

Proof of Lemma

Want to show: AC_H does not derive the empty list on G iff $G \to P(H)$.

'$\Rightarrow$'
Let $L(x)$ be the (non-empty) set at the final stage of AC_H.
Claim: $x \mapsto L(x)$ is a homomorphism from G to $P(H)$.
Suppose $(a, b) \in E(G)$. By Definition of AC_H:
for each $u \in L(a)$ there is $v \in L(b)$ such that $(u, v) \in E(H)$.
for each $v \in L(b)$ there is $u \in L(a)$ such that $(u, v) \in E(H)$.
By definition of $P(H)$, we have $(L(a), L(b)) \in E(P(H))$.

'$\Leftarrow$'
Let $f \colon G \to P(H)$ a homomorphism.
Then AC_H doesn't remove values from $f(x)$ from $L(x)$:

Proof of Lemma

Want to show: AC_H does not derive the empty list on G iff $G \to P(H)$.

'$\Rightarrow$'

Let $L(x)$ be the (non-empty) set at the final stage of AC_H.
Claim: $x \mapsto L(x)$ is a homomorphism from G to $P(H)$.
Suppose $(a, b) \in E(G)$. By Definition of AC_H:
for each $u \in L(a)$ there is $v \in L(b)$ such that $(u, v) \in E(H)$.
for each $v \in L(b)$ there is $u \in L(a)$ such that $(u, v) \in E(H)$.
By definition of $P(H)$, we have $(L(a), L(b)) \in E(P(H))$.

'$\Leftarrow$'

Let $f \colon G \to P(H)$ a homomorphism.
Then AC_H doesn't remove values from $f(x)$ from $L(x)$:
Suppose $(a, b) \in E(G)$.

Proof of Lemma

Want to show: AC_H does not derive the empty list on G iff $G \to P(H)$.

'$\Rightarrow$'

Let $L(x)$ be the (non-empty) set at the final stage of AC_H.
Claim: $x \mapsto L(x)$ is a homomorphism from G to $P(H)$.
Suppose $(a, b) \in E(G)$. By Definition of AC_H:
for each $u \in L(a)$ there is $v \in L(b)$ such that $(u, v) \in E(H)$.
for each $v \in L(b)$ there is $u \in L(a)$ such that $(u, v) \in E(H)$.
By definition of $P(H)$, we have $(L(a), L(b)) \in E(P(H))$.

'$\Leftarrow$'

Let $f \colon G \to P(H)$ a homomorphism.
Then AC_H doesn't remove values from $f(x)$ from $L(x)$:
Suppose $(a, b) \in E(G)$. For each $u \in f(a)$ there is $v \in f(b) \subseteq L(b)$ such that $(u, v) \in E(H)$.

Proof of Lemma

Want to show: AC_H does not derive the empty list on G iff $G \to P(H)$.

'$\Rightarrow$'

Let $L(x)$ be the (non-empty) set at the final stage of AC_H.
Claim: $x \mapsto L(x)$ is a homomorphism from G to $P(H)$.
Suppose $(a, b) \in E(G)$. By Definition of AC_H:
for each $u \in L(a)$ there is $v \in L(b)$ such that $(u, v) \in E(H)$.
for each $v \in L(b)$ there is $u \in L(a)$ such that $(u, v) \in E(H)$.
By definition of $P(H)$, we have $(L(a), L(b)) \in E(P(H))$.

'$\Leftarrow$'

Let $f \colon G \to P(H)$ a homomorphism.
Then AC_H doesn't remove values from $f(x)$ from $L(x)$:
Suppose $(a, b) \in E(G)$. For each $u \in f(a)$ there is $v \in f(b) \subseteq L(b)$ such that $(u, v) \in E(H)$. Hence, u is not removed from $L(a)$.

Proof

Want to show: AC solves CSP(H) iff $P(H) \rightarrow H$.

Proof

Want to show: AC solves CSP(H) iff $P(H) \to H$.

'$\Rightarrow$'
Suppose AC solves CSP(H).

Proof

Want to show: AC solves CSP(H) iff $P(H) \to H$.

'$\Rightarrow$'
Suppose AC solves CSP(H).
Apply AC_H to $P(H)$.

Proof

Want to show: AC solves CSP(H) iff $P(H) \to H$.

'$\Rightarrow$'
Suppose AC solves CSP(H).
Apply AC_H to $P(H)$.
Since $H \to P(H)$, previous lemma shows: AC does not derive empty list.

Proof

Want to show: AC solves CSP(H) iff $P(H) \to H$.

'$\Rightarrow$'
Suppose AC solves CSP(H).
Apply AC_H to $P(H)$.
Since $H \to P(H)$, previous lemma shows: AC does not derive empty list.
Hence $P(H) \to H$.

Proof

Want to show: AC solves CSP(H) iff $P(H) \to H$.

'$\Rightarrow$'

Suppose AC solves CSP(H).

Apply AC$_H$ to $P(H)$.

Since $H \to P(H)$, previous lemma shows: AC does not derive empty list.

Hence $P(H) \to H$.

'$\Leftarrow$'

Proof

Want to show: AC solves CSP(H) iff $P(H) \to H$.

'$\Rightarrow$'
Suppose AC solves CSP(H).
Apply AC$_H$ to $P(H)$.
Since $H \to P(H)$, previous lemma shows: AC does not derive empty list.
Hence $P(H) \to H$.

'$\Leftarrow$'
Suppose $P(H) \to H$.

Proof

Want to show: AC solves CSP(H) iff $P(H) \to H$.

'$\Rightarrow$'
Suppose AC solves CSP(H).
Apply AC_H to $P(H)$.
Since $H \to P(H)$, previous lemma shows: AC does not derive empty list.
Hence $P(H) \to H$.

'$\Leftarrow$'
Suppose $P(H) \to H$.

- If AC derives empty list on G, by Lemma $G \not\to P(H)$.

Proof

Want to show: AC solves CSP(H) iff $P(H) \rightarrow H$.

'$\Rightarrow$'
Suppose AC solves CSP(H).
Apply AC_H to $P(H)$.
Since $H \rightarrow P(H)$, previous lemma shows: AC does not derive empty list.
Hence $P(H) \rightarrow H$.

'$\Leftarrow$'
Suppose $P(H) \rightarrow H$.

- If AC derives empty list on G, by Lemma $G \not\rightarrow P(H)$.
  Since $H \rightarrow P(H)$, we conclude that $G \not\rightarrow H$.

Proof

Want to show: AC solves CSP(H) iff $P(H) \to H$.

'$\Rightarrow$'
Suppose AC solves CSP(H).
Apply AC_H to $P(H)$.
Since $H \to P(H)$, previous lemma shows: AC does not derive empty list.
Hence $P(H) \to H$.

'$\Leftarrow$'
Suppose $P(H) \to H$.

- If AC derives empty list on G, by Lemma $G \not\to P(H)$.
  Since $H \to P(H)$, we conclude that $G \not\to H$.
- If AC does not derive the empty list on G, by Lemma $G \to P(H)$.

Proof

Want to show: AC solves CSP(H) iff $P(H) \rightarrow H$.

'$\Rightarrow$'

Suppose AC solves CSP(H).
Apply AC_H to $P(H)$.
Since $H \rightarrow P(H)$, previous lemma shows: AC does not derive empty list.
Hence $P(H) \rightarrow H$.

'$\Leftarrow$'

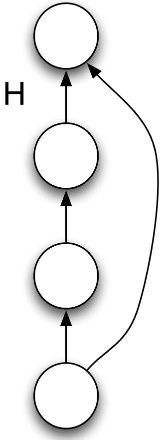
Suppose $P(H) \rightarrow H$.

- If AC derives empty list on G, by Lemma $G \not\rightarrow P(H)$.
  Since $H \rightarrow P(H)$, we conclude that $G \not\rightarrow H$.
- If AC does not derive the empty list on G, by Lemma $G \rightarrow P(H)$.
  Composing homomorphisms, we obtain $G \rightarrow H$.

QED.

Exercise

Does Arc Consistency solve CSP(H) for the following graph H:



Tree Duality

Facts.

For all digraphs G,

$$G \to \vec{P}_2 \quad \Leftrightarrow \quad Z_1, Z_2, \dots \not\to G$$

CSP$(\vec{P}_2)$ can be solved by AC.

Tree Duality

Facts.

For all digraphs G,

$$G \to \vec{P}_2 \quad \Leftrightarrow \quad Z_1, Z_2, \dots \not\to G$$

CSP$(\vec{P}_2)$ can be solved by AC.

For all digraphs G,

$$G \to T_3 \quad \Leftrightarrow \quad \vec{P}_3 \not\to G$$

CSP(T_3) can be solved by AC.

Tree Duality

Facts.

For all digraphs G,

$$G \to \vec{P}_2 \quad \Leftrightarrow \quad Z_1, Z_2, \cdots \not\to G$$

CSP($\vec{P}_2$) can be solved by AC.

For all digraphs G,

$$G \to T_3 \quad \Leftrightarrow \quad \vec{P}_3 \not\to G$$

CSP(T_3) can be solved by AC.

Definition

A digraph H has *tree duality* if there is a (not necessarily finite) set $\mathcal{N}$ of orientations of trees such that

$$\forall G: \quad G \to H \quad \Leftrightarrow \quad \mathcal{N} \not\to G$$

Tree Duality, Power Graph, Arc-Consistency

Theorem 2.

Let H be a finite digraph. The following are equivalent.

1. AC solves CSP(H)
2. $P(H)$ homomorphically maps to H
3. H has tree duality.

Tree Duality, Power Graph, Arc-Consistency

Theorem 2.

Let H be a finite digraph. The following are equivalent.

1. AC solves CSP(H)
2. $P(H)$ homomorphically maps to H
3. H has tree duality.

Already saw $1 \Leftrightarrow 2$.

Tree Duality, Power Graph, Arc-Consistency

Theorem 2.

Let H be a finite digraph. The following are equivalent.

1. AC solves CSP(H)
2. $P(H)$ homomorphically maps to H
3. H has tree duality.

Already saw $1 \Leftrightarrow 2$.

Proof ideas for $3 \Rightarrow 2$ and $1 \Rightarrow 3$ (details in script)

Tree Duality, Power Graph, Arc-Consistency

Theorem 2.

Let H be a finite digraph. The following are equivalent.

1. AC solves CSP(H)
2. $P(H)$ homomorphically maps to H
3. H has tree duality.

Already saw $1 \Leftrightarrow 2$.

Proof ideas for $3 \Rightarrow 2$ and $1 \Rightarrow 3$ (details in script)

$3 \Rightarrow 2$:

Tree Duality, Power Graph, Arc-Consistency

Theorem 2.

Let H be a finite digraph. The following are equivalent.

1. AC solves CSP(H)
2. $P(H)$ homomorphically maps to H
3. H has tree duality.

Already saw $1 \Leftrightarrow 2$.

Proof ideas for $3 \Rightarrow 2$ and $1 \Rightarrow 3$ (details in script)

$3 \Rightarrow 2$: suffices to show that for all trees T,

$$T \to P(H) \quad \Rightarrow \quad T \to H$$

Theorem 2.

Let H be a finite digraph. The following are equivalent.

1. AC solves CSP(H)
2. $P(H)$ homomorphically maps to H
3. H has tree duality.

Already saw 1 $\Leftrightarrow$ 2.
Proof ideas for 3 $\Rightarrow$ 2 and 1 $\Rightarrow$ 3 (details in script)
3 $\Rightarrow$ 2: suffices to show that for all trees T,

$$T \to P(H) \quad \Rightarrow \quad T \to H$$

1 $\Rightarrow$ 3:

Tree Duality, Power Graph, Arc-Consistency

Theorem 2.

Let H be a finite digraph. The following are equivalent.

1. AC solves CSP(H)
2. $P(H)$ homomorphically maps to H
3. H has tree duality.

Already saw 1 $\Leftrightarrow$ 2.
Proof ideas for 3 $\Rightarrow$ 2 and 1 $\Rightarrow$ 3 (details in script)
3 $\Rightarrow$ 2: suffices to show that for all trees T,

$$T \to P(H) \quad \Rightarrow \quad T \to H$$

1 $\Rightarrow$ 3: If AC derives the empty list on G, then the derivation has a tree-like structure.

Concluding Remarks

Concluding Remarks

- Arc Consistency can be generalized to general relational structures (relations of arbitrary arity, arbitrarily many relations).

Concluding Remarks

- Arc Consistency can be generalized to general relational structures (relations of arbitrary arity, arbitrarily many relations).
- Power-set graph can be generalized to general relational structures.

Concluding Remarks

- Arc Consistency can be generalized to general relational structures (relations of arbitrary arity, arbitrarily many relations).
- Power-set graph can be generalized to general relational structures.
- Today's proofs still work.

Concluding Remarks

- Arc Consistency can be generalized to general relational structures (relations of arbitrary arity, arbitrarily many relations).
- Power-set graph can be generalized to general relational structures.
- Today's proofs still work.
- In this more general setting, Boolean structures preserved by min or by max can be solved by Arc-Consistency.