A logical approach to Isomorphism Testing and Constraint Satisfaction

Oleg Verbitsky

Humboldt University of Berlin, Germany

ESSLLI 2016, 15–19 August

Part 6:
Existential-positive FO^2 and Constraint Satisfaction

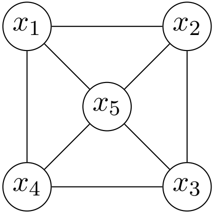
Outline

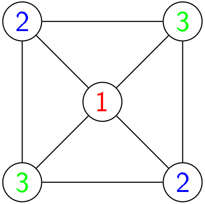
1. Constraint Satisfaction Problem and Constraint Propagation
2. The existential k-pebble game
3. k-Consistency Checking
4. Time complexity of Arc Consistency

Outline

Constraint Satisfaction Problem (CSP)

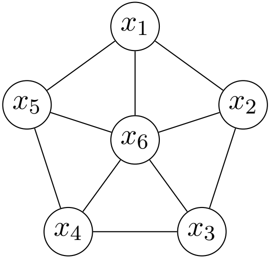
Variables	x_1, x_2, x_3, x_4, x_5
Values	$x_i \in \{1, 2, 3\}$
Constraints	$x_1 \neq x_2$, $x_2 \neq x_3$, $x_3 \neq x_4$, $x_4 \neq x_1$, $x_1 \neq x_5$, $x_2 \neq x_5$, $x_3 \neq x_5$, $x_4 \neq x_5$
Question:	Is there an assignment of values to the variables satisfying all constraints?





Constraint Satisfaction Problem (CSP)

Variables	$x_1, x_2, x_3, x_4, x_5, x_6$
Values	$x_i \in \{1, 2, 3\}$
Constraints	$x_1 \neq x_2$, $x_2 \neq x_3$, $x_3 \neq x_4$, $x_4 \neq x_5$, $x_5 \neq x_1$, $x_1 \neq x_6$, $x_2 \neq x_6$, $x_3 \neq x_6$, $x_4 \neq x_6$, $x_5 \neq x_6$
Question:	Is there an assignment of values to the variables satisfying all constraints?

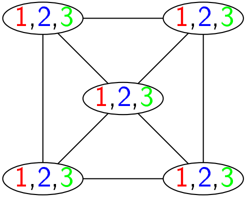


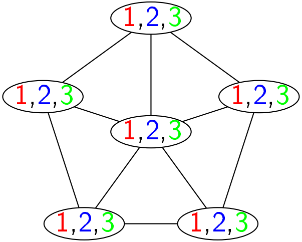
No!

Methodolody: derivation instead of search

Example: 3-COLORABILITY.
We can choose an arbitrary edge and color it arbitrarily.

Derivation rules: $\dfrac{x = 1,\ y \neq x}{y \neq 1}$ etc.

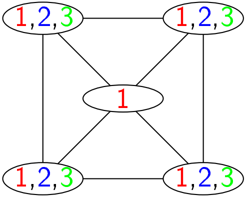


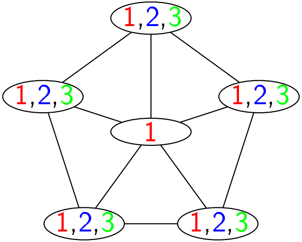


Methodolody: derivation instead of search

Example: 3-COLORABILITY.
We can choose an arbitrary edge and color it arbitrarily.

Derivation rules: $\dfrac{x = 1,\ \ y \neq x}{y \neq 1}$ etc.



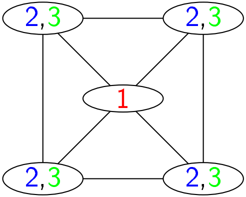


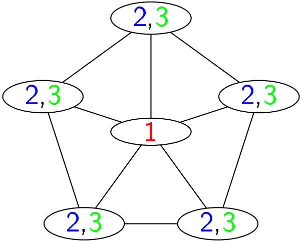
Constraint propagation

Methodolody: derivation instead of search

Example: 3-COLORABILITY.
We can choose an arbitrary edge and color it arbitrarily.

Derivation rules: $\dfrac{x = 1,\ y \neq x}{y \neq 1}$ etc.

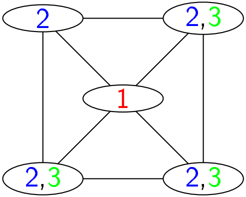


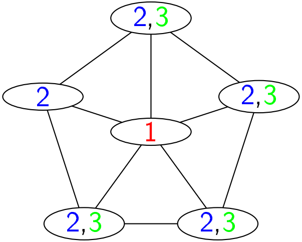


Methodolody: derivation instead of search

Example: 3-COLORABILITY.
We can choose an arbitrary edge and color it arbitrarily.

Derivation rules: $\dfrac{x = 1,\ y \neq x}{y \neq 1}$ etc.

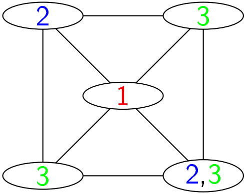


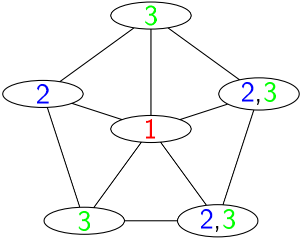


Methodolody: derivation instead of search

Example: 3-COLORABILITY.
We can choose an arbitrary edge and color it arbitrarily.

Derivation rules: $\dfrac{x = 1,\ y \neq x}{y \neq 1}$ etc.



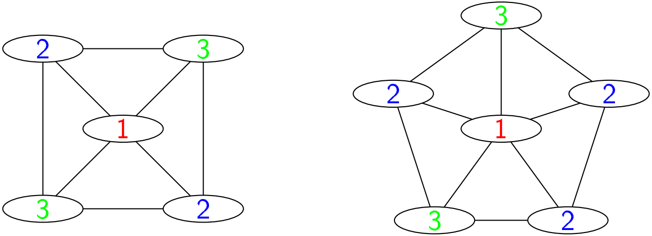


Constraint propagation

Methodolody: derivation instead of search

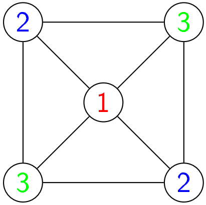
Example: 3-COLORABILITY.
We can choose an arbitrary edge and color it arbitrarily.

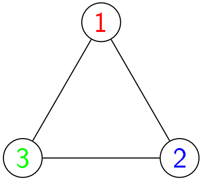
Derivation rules: $\dfrac{x = 1,\ y \neq x}{y \neq 1}$ etc.



The Feder-Vardi paradigm: a CSP = a Homomorphism Problem

For example, a graph G is 3-colorable iff there is a homomorphism from G to K_3 (notation: $G \to K_3$).





Outline

A logic and a game for the Homomorphism Problem

The following three conditions are equivalent:

- $G \not\to H$,
- some existential-positive formula distinguishes G from H,
- Spoiler has a winning strategy in the existential k-pebble game on G and H for some k.

A logic and a game for the Homomorphism Problem

The following three conditions are equivalent:

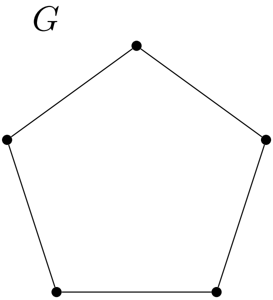
- $G \not\to H$,
- some existential-positive formula distinguishes G from H,
- Spoiler has a winning strategy in the existential k-pebble game on G and H for some k.

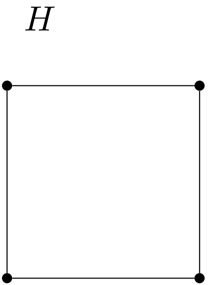
The existential-positive logic $\mathrm{FO}_{\exists,+}$ allows only monotone Boolean connectives (no negation) and only existential quantifiers (no universal quantification).

The existential k-pebble game on G and H is the version of the k-pebble Ehrenfeucht game where

- Spoiler moves always in G,
- Duplicator must keep a partial homomorphism.

The existential k-pebble game (Kolaitis-Vardi)





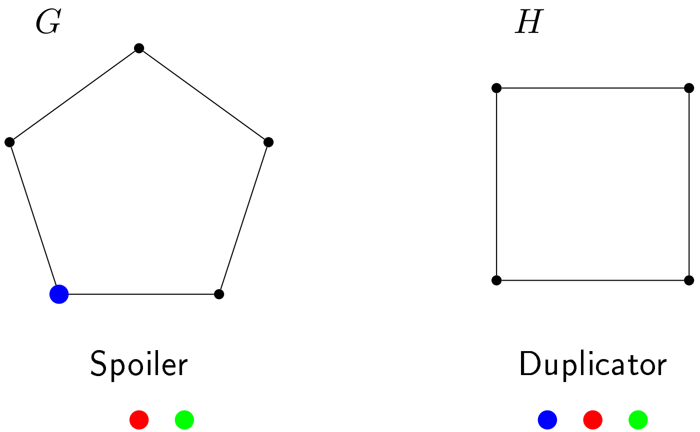
Spoiler

Objective:
show that $\neg\exists$ homom. $G \to H$

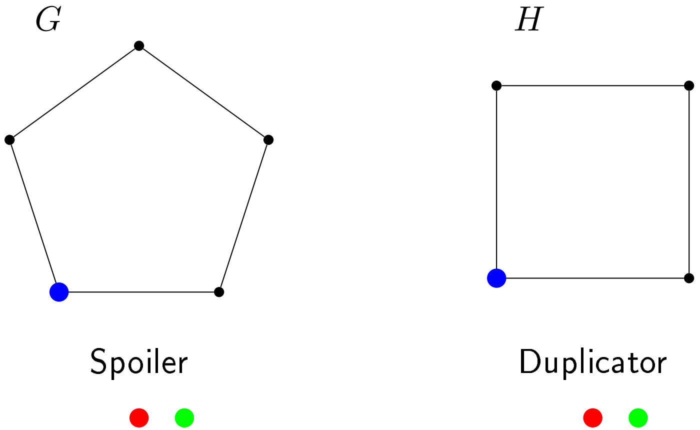
Duplicator

it may exist

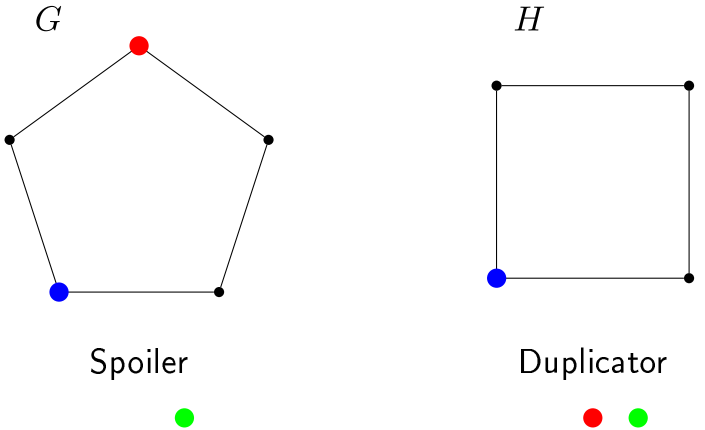
The existential k-pebble game (Kolaitis-Vardi)



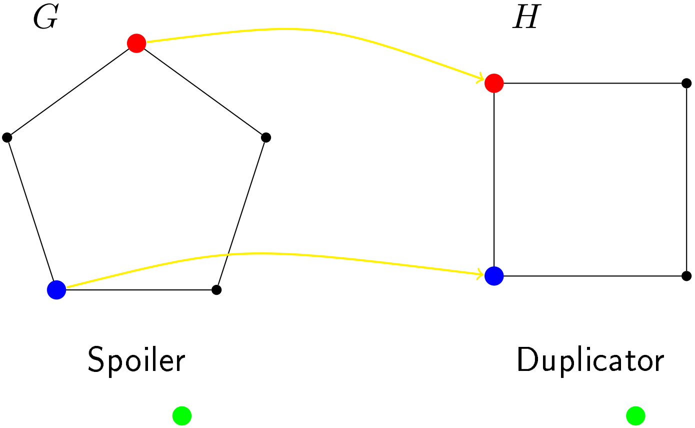
The existential k-pebble game (Kolaitis-Vardi)



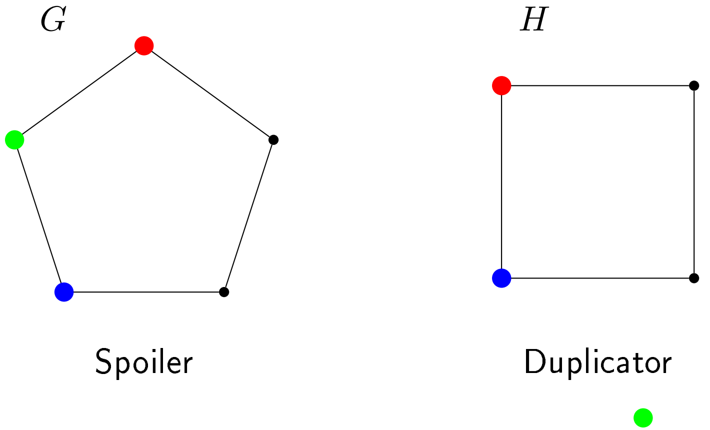
The existential k-pebble game (Kolaitis-Vardi)



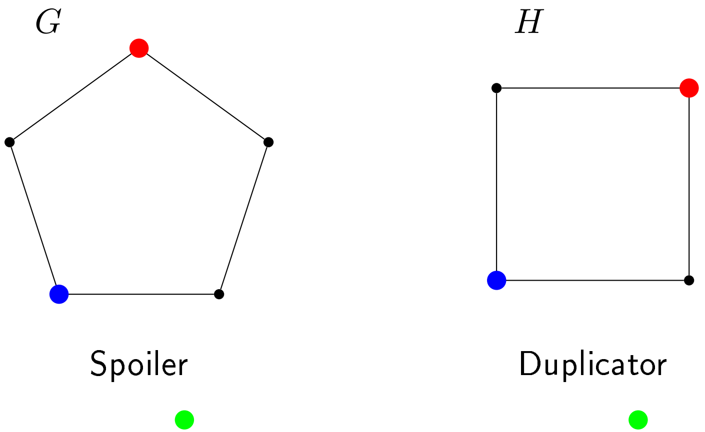
The existential k-pebble game (Kolaitis-Vardi)



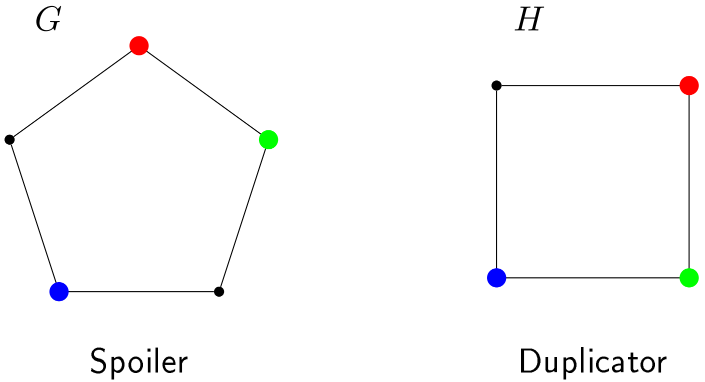
The existential k-pebble game (Kolaitis-Vardi)



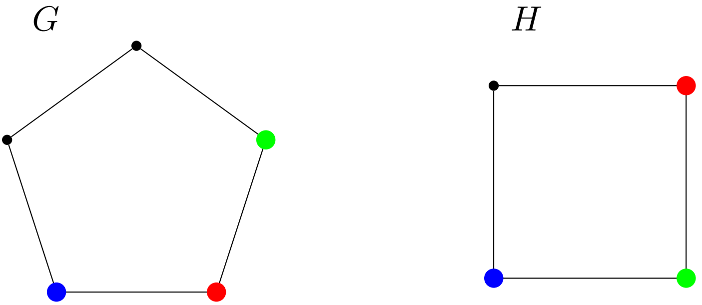
The existential k-pebble game (Kolaitis-Vardi)



The existential k-pebble game (Kolaitis-Vardi)



The existential k-pebble game (Kolaitis-Vardi)



Spoiler

Duplicator

The existential k-pebble game

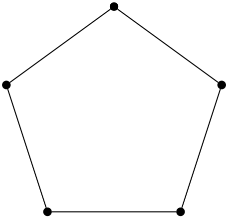
Observation:

If there is a homomorphism h from G to H, then Duplicator wins by pebbling $h(v)$ if Spoiler pebbles v.

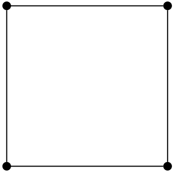
Hence, the game can be used as heuristics for the CSP.

Of course, it is incomplete if k is not large enough.
In our example, $k = 3$ is enough and $k = 2$ is not.

G

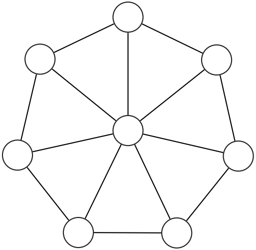


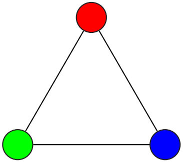
H



Another example

Let G_n denote the wheel graph with n vertices.
If n is even, then Spoiler wins the existential game on G_n and K_3 with 4 pebbles.





Spoiler

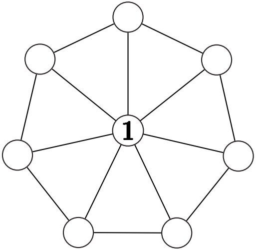
1 2 3 4

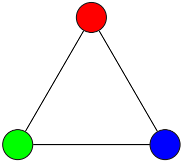
Duplicator

1 2 3 4

Another example

Let G_n denote the wheel graph with n vertices.
If n is even, then Spoiler wins the existential game on G_n and K_3 with 4 pebbles.



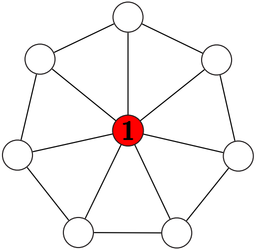


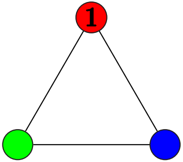
Spoiler
2 3 4

Duplicator
1 2 3 4

Another example

Let G_n denote the wheel graph with n vertices.
If n is even, then Spoiler wins the existential game on G_n and K_3 with 4 pebbles.





Spoiler

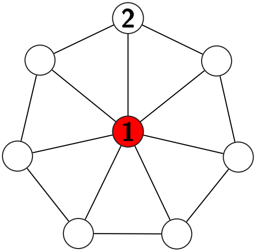
2 3 4

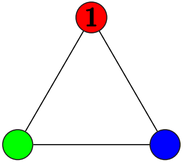
Duplicator

2 3 4

Another example

Let G_n denote the wheel graph with n vertices.
If n is even, then Spoiler wins the existential game on G_n and K_3 with 4 pebbles.





Spoiler

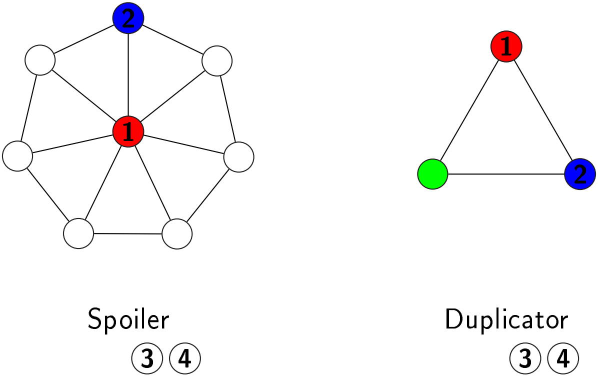
3 4

Duplicator

2 3 4

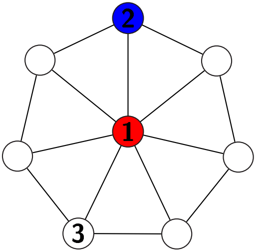
Another example

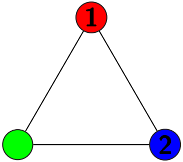
Let G_n denote the wheel graph with n vertices.
If n is even, then Spoiler wins the existential game on G_n and K_3 with 4 pebbles.



Another example

Let G_n denote the wheel graph with n vertices.
If n is even, then Spoiler wins the existential game on G_n and K_3 with 4 pebbles.





Spoiler

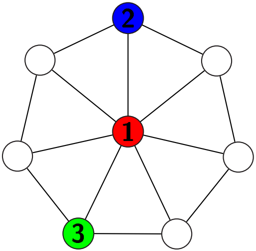
4

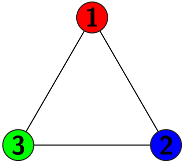
Duplicator

3 4

Another example

Let G_n denote the wheel graph with n vertices.
If n is even, then Spoiler wins the existential game on G_n and K_3 with 4 pebbles.





Spoiler

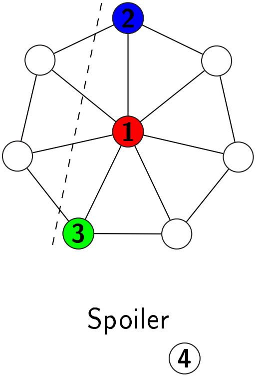
4

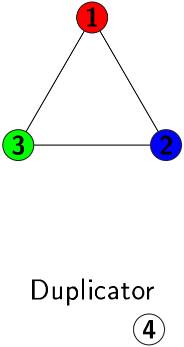
Duplicator

4

Another example

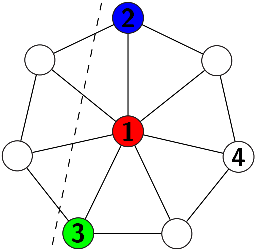
Let G_n denote the wheel graph with n vertices.
If n is even, then Spoiler wins the existential game on G_n and K_3 with 4 pebbles.



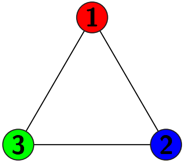


Another example

Let G_n denote the wheel graph with n vertices.
If n is even, then Spoiler wins the existential game on G_n and K_3 with 4 pebbles.



Spoiler

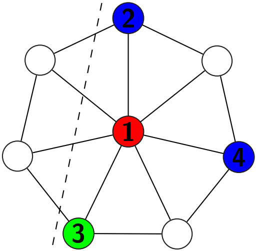


Duplicator

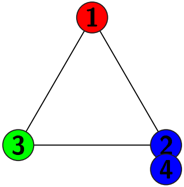
4

Another example

Let G_n denote the wheel graph with n vertices.
If n is even, then Spoiler wins the existential game on G_n and K_3 with 4 pebbles.



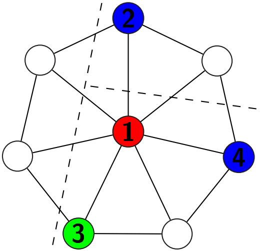
Spoiler



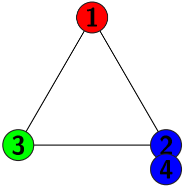
Duplicator

Another example

Let G_n denote the wheel graph with n vertices.
If n is even, then Spoiler wins the existential game on G_n and K_3 with 4 pebbles.



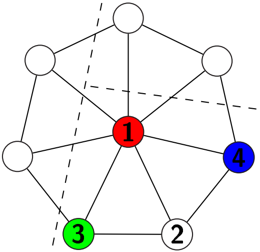
Spoiler



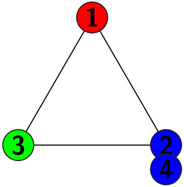
Duplicator

Another example

Let G_n denote the wheel graph with n vertices.
If n is even, then Spoiler wins the existential game on G_n and K_3 with 4 pebbles.



Spoiler



Duplicator

Outline

The triad: A logic, a game, an algorithm

Theorem (Kolaitis, Vardi 95)

Suppose that $G \not\to H$. Then the following three conditions are equivalent:

- $W_{\exists,+}(G,H) \leq k$, i.e., G is distinguishable from H by an existential-positive sentence with k variables;
- Spoiler wins the existential k-pebble game on G and H;
- k-Consistency Checking recognizes that $G \not\to H$.

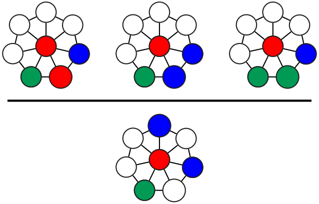
k-Consistency Checking (recasted)

Algorithmic problem

Given two finite structures G and H, does Spoiler win the existential k-pebble game on these structures?

- This is a relaxation of the homomorphism problem.
- For small k, it is commonly used as a heuristics approach.

A propagation-based algorithm makes derivations like



- winning positions for Spoiler

$\Downarrow$

- a winning position too

(a position is a mapping of $\leq k$ vertices from $V(G)$ into $V(H)$)

Spoiler has a winning stategy $\Leftrightarrow$ the uncolored graph is derivable.

The time complexity of k-Consistency Checking

Since there are at most $N = v(G)^k v(H)^k$ positions, all derivations can be generated in time N^{k+1} (the wasteful version of k-consistency checking).

Nevertheless, if k is fixed, this takes polynomial time (while CSP is NP-complete).

The time complexity of k-Consistency Checking

Theorem

The k-Consistency problem is solvable in

- *time $O(v(G)^k v(H)^k) = O(n^{2k})$ for each k* *[Cooper 89]*
- *but not in time $O(n^{\frac{k-3}{12}})$ for $k \geq 15$* *[Berkholz 12]*

The time complexity of k-Consistency Checking

Theorem

The k-Consistency problem is solvable in

- *time* $O(v(G)^k v(H)^k) = O(n^{2k})$ *for each* k *[Cooper 89]*
- *but not in time* $O(n^{\frac{k-3}{12}})$ *for* $k \geq 15$ *[Berkholz 12]*

Question. What about Arc Consistency ($k = 2$)?

Remark. If $k = 2$, we consider directed graphs.

The time complexity of k-Consistency Checking

Theorem

The k-Consistency problem is solvable in

- *time* $O(v(G)^k v(H)^k) = O(n^{2k})$ *for each* k [Cooper 89]
- *but not in time* $O(n^{\frac{k-3}{12}})$ *for* $k \geq 15$ [Berkholz 12]

Question. What about Arc Consistency $(k = 2)$?

Remark. If $k = 2$, we consider directed graphs.

In practice: All known Arc Consistency $(k = 2)$ algorithms

- AC-1, AC-3, AC-3.1 / AC-2001, AC-3.2, AC-3.3, AC-3$_d$, AC-4, AC-5, AC-6, AC-7, AC-8, AC-$*$ etc.

are based on constraint propagation.

Outline

Bounds for the propagation approach

Denote the number of vertices and edges by $v(G)$ and $e(G)$ resp.

Theorem (Berkholz, V. 2013)

1. *Arc Consistency is solvable in time* $O(v(G)e(H) + e(G)v(H))$, *which implies* $O(n^3)$ *in terms of* $n = v(G) + v(H)$.
2. *Any propagation-based arc consistency algorithm takes time* $\Omega(n^3)$.

Proof-scheme

- A propagation-based algorithm $\mapsto$ a winning strategy for Spoiler
- The time on input $G, H \mapsto$ the size of the game tree
- In fact, it is enough to show that the optimum depth of the game tree, which is equal to $D_{\exists,+}(G, H)$, is large for some G and H.

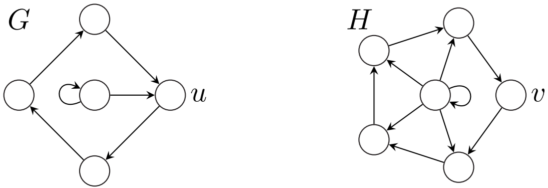
Lemma (constraint propagation can be slow)

There are directed graphs G and H with $v(G) = v(H) - 1 = n$ such that

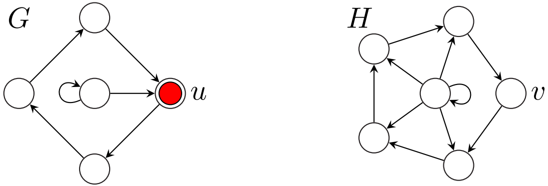
- Spoiler wins the existential 2-pebble game on G and H;
- Duplicator can resist in $\Omega(n^2)$ rounds.

Remark. $n^2 + 1$ rounds always suffice for Spoiler.

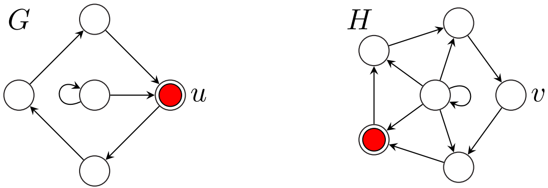
An example of slow constraint propagation



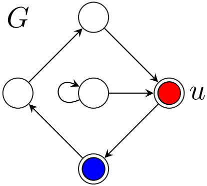
An example of slow constraint propagation

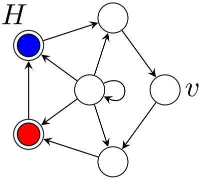


An example of slow constraint propagation

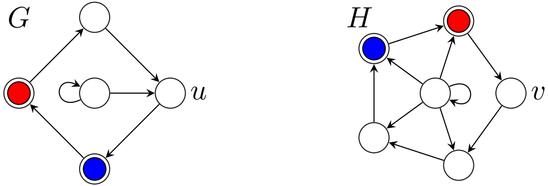


An example of slow constraint propagation

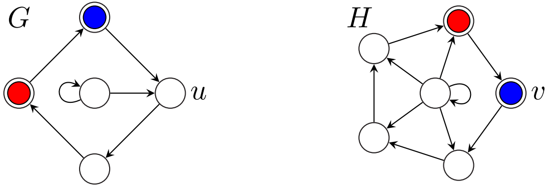




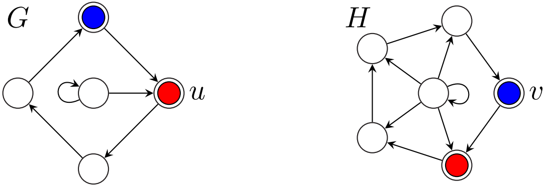
An example of slow constraint propagation



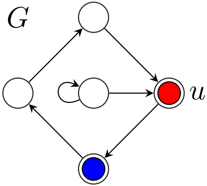
An example of slow constraint propagation

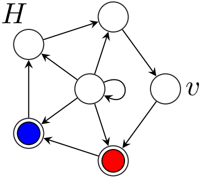


An example of slow constraint propagation

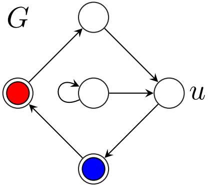


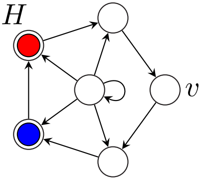
An example of slow constraint propagation



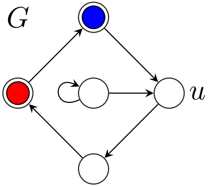


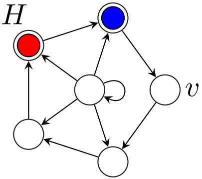
An example of slow constraint propagation



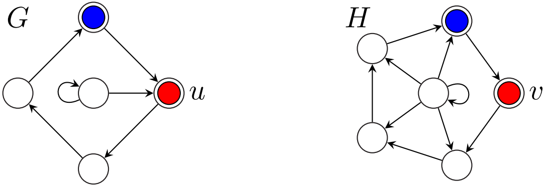


An example of slow constraint propagation

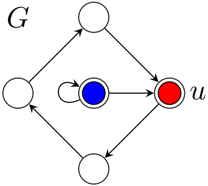


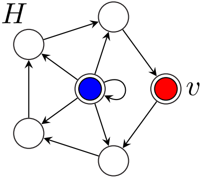


An example of slow constraint propagation



An example of slow constraint propagation





Open problem

Can Arc Consistency be solved faster than in time $O(n^3)$ (by methods different from constraint propagation)?

References

- T. Feder, M. Vardi. The computational structure of monotone monadic SNP and constraint satisfaction: A study through datalog and group theory. SIAM Journal on Computing 28:57–104 (1998).
- C. Berkholz. Lower bounds for existential pebble games and k-consistency tests. LICS'12.
- C. Berkholz, O. Verbitsky: On the speed of Constraint Propagation and the time complexity of Arc Consistency testing. MFCS'13.