Definability and Computability

Final Project:

The expression of graph properties in subsets of second order logic.

Ran Wolff 2846998-9

Irit Katriel 2973179-1

1 Introduction

This essay deals with the mapping of a number of graph properties into subsets of second-order logic (SOL) in which they can be expressed. Such a subset of SOL is defined by the number of alternations of the second-order quantifiers in an SOL that defines the property, as well as the arity of the sets that are quantified. The intention of this project is to serve as a road-map and reference for the computability and definability course.

The SOL definitions we use are over the structure known as GRAPH1, which includes the vertices of the graph as a set and its edges as a 2-ary relation over the set of vertices. Formally: $GRAPH1 =< V^1 = \{1, \cdots, n\}, E^2 \subseteq V \times V >$

Our work calls for much further development. In some places we gave pointers to questions we do not answer here.

Structure of this essay:

First we provide a summary of the results we have arrived at in a table that maps graph properties to subsets of SOL that are defined by the arity and alternation of the SOL quantifiers in the simplest SOL formula we could find.

In section 2 we define (verbally) the graph properties that are considered in this work.

In section 3 we provide a list of "utilities". That is, we provide a set of sub-problems over graphs and spell out formulas that define them and can later be used as sub-formulas for our SOL definitions of the graph properties that are listed in section 2.

In section 4 we prove the mapping of our selected set of graph properties into the table, by writing an SOL formula that for each such property, showing its containment in the subset of SOL to which it was mapped.

2 Definitions of the graph properties in our table:

Connectivity:

Given a graph G, the property of Connectivity is satisfied iff for every two vertices a,b in G, there exists a path between a and b.

D-Connectivity:

Same as Connectivity, over directed graphs.

Non-Connectivity:

Given a graph G, the property of Non-Connectivity is satisfied iff the property of Connectivity is not satisfied.

D-Non-Connectivity:

Same as Non-Connectivity, over directed graphs.

Reachability(x,y):

Given a graph G and two vertices x,y in G, the property of Reachability is satisfied iff there is a path between x and y.

Un-Reachability(x,y):

Given a graph G and two vertices x,y in G, the property of Un-Reachability(x,y) is satisfied iff the property of Reachability(x,y) is not satisfied.

D-Reachability(x,y):

Given a directed graph G and two vertices x,y in G, the property of D-Reachability(x,y) is satisfied iff there is a directed path from x to y in G.

D-Un-Reachability(x,y):

Given a directed graph G and two vertices x,y in G, the property of D-Un-Reachability(x,y) is satisfied iff the property of D-Reachability(x,y) is not satisfied.

k-Seperability(a,b):

Given a graph G and two vertices a,b the property of k-Seperability(a,b) is satisfied iff a and b can be separated by eliminating at most k vertices in the graph.

Non-k-Seperability(a,b):

given a graph G and two vertices a,b the property of Non-k-Seperability(a,b) is satisfied iff the property of k-Seperability(a,b) is not satisfied.

Tree:

Given a graph G, the property of Tree is satisfied iff the graph G is a tree.

Non-Tree:

Given a graph G, the property of Non-Tree is satisfied iff the property of Tree is not satisfied.

3-Colorability:

Given a graph G, the property of 3-Colorability is satisfied iff there exists a coloring of the vertices of G with three colors such that no two adjacent vertices are colored in the same color.

Non-3-Colorability: Given a graph G, the property of Non-3-Colorability is satisfied iff the property of 3-Colorability is not satisfied.

Chordality:

Given a graph G, the property of Chordality is satisfied iff G has no induces sub-graph which is a cycle with at least 4 vertices.

Non-Chordality:

Given a graph G, the property of Non-Chordality is satisfied iff the property of Non-Chordality is not satisfied.

k-Disjoint-Paths($x_1, x_2 \cdots x_k; y_1, y_2 \cdots y_k$):

Given a graph G, the property of k-Disjoint-Paths($x_1, x_2 \cdots x_k; y_1, y_2 \cdots y_k$) is satisfied iff $x_1 \neq y_1, x_2 \neq y_2, \cdots, x_k \neq y_k$ and there exist k paths $p_1, \cdots, p_k$ linking x_1 to y_1, x_2 to $y_2, \cdots, x_k$ to y_k respectively, such that every vertex common to two paths p_i and p_j, $j \neq i$, belongs to $x_i, y_i \cap x_j, y_j$.

Non-k-Disjoint-Paths($x_1, x_2 \cdots x_k; y_1, y_2 \cdots y_k$):

Given a graph G, the property of Non-k-Disjoint-Paths($x_1, x_2 \cdots x_k; y_1, y_2 \cdots y_k$) is satisfied iff the property of k-Disjoint-Paths($x_1, x_2 \cdots x_k; y_1, y_2 \cdots y_k$) is not satisfied.

3 Some utilities:

In this section we define some formulas that we will later use as building blocks for constructing definitions for our selected graph properties. For each such sub-problem, the problem is first defines as a graph property. Then, an alternative, *verbal* definition, which is easy to be translated into a formula, is provided and proved to be equivalent to the original definition. Lastly, a formula defining the property is given.

3-Cycle:

Problem definition: Given a graph G, the property 3-Cycle is satisfied iff there is a cycle of length exactly 3 in G.

Verbal definition: The graph G satisfies 3-Cycle iff there exist three distinct vertices such that there is an edge between every two of them.

Proof:

There is a cycle of size exactly 3 in a graph $\Leftrightarrow$ there are three vertices that form a cycle $\Leftrightarrow$ there are three vertices that are distinct and there is an edge between every two of them.

Formula:

$\Psi_{3-cycle} = \exists x \exists y \exists z [(x \neq y) \wedge (y \neq z) \wedge (x \neq z) \wedge E(x,y) \wedge E(y,z) \wedge E(x,z)]$

Cycle:

Problem definition:

Given a graph G, the property Cycle is satisfied iff there is a cycle in G.

Verbal definition:

The graph G satisfies Cycle iff there are two vertices x and y in G such that there are two disjoint paths between x and y.

Proof:

$\Rightarrow$: The graph G satisfies Cycle $\Rightarrow$ there is a cycle in G $\Rightarrow$ Let u,v be two distinct vertices in the cycle $\Rightarrow$ there are two disjoint paths between u and v, which together form the cycle.

$\Leftarrow$: There exist two disjoint vertices x and y in G such that there are two disjoint paths between x and y $\Rightarrow$ The two disjoint paths add up into a cycle $\Rightarrow$ There is a cycle in G $\Rightarrow$ G satisfies Cycle.

Formula:

$\Psi_{Cycle} = \exists x \exists y \Psi_{2-Disjoint-Paths}(x, y)$

(See the definition of k-Disjoint-Paths below)

<u>3-Disjoint-Sets(X,Y,Z)</u>

Problem definition:

Three sets X, Y and Z satisfy 3-Disjoint-Sets(X,Y,Z) iff they are pairwise disjoint.

Verbal definition:

The three sets satisfy 3-Disjoint-Sets iff there does not exist and element that appears in two of them.

Proof:

The three sets satisfy 3-Disjoint-Sets $\Leftrightarrow$ they are pairwise disjoint $\Leftrightarrow$ no two of them have an element in common.

Formula:

$\Psi_{3-Disjoint-Sets}(X, Y, Z) = \neg\exists x((x \in X \wedge x \in Y) \vee (x \in Y \wedge x \in Z) \vee (x \in X \wedge x \in Z))$

<u>Edge-Closed(X)</u>

Problem definition:

Given a graph G and a subset X of the vertices of G, X satisfies Edge-Closed if there is no edge in G that connects a vertex that is in X with a vertex that is not in X.

Verbal definition:

X satisfies Edge-Closed iff there do not exist two vertices x, y such that x is in X, y is not in X and there is an edge between x and y.

Proof:

By definition.

Formula:

$\Psi_{Edge-Closed}(X) = \neg\exists x \exists y[(x \in X) \wedge (y \notin X) \wedge (E(x, y) \vee E(y, x))]$

<u>D-Edge-Closed(X)</u>

Problem definition:

Given a directed graph G and a subset X of the vertices of G, X satisfies D-Edge-Closed iff there is no edge in G that is directed from a vertex in X to a vertex that is not in X.

Verbal definition:

X satisfies D-Edge-Closed iff there do not exist two vertices x and y such that x is in X, y is not in X and there is an edge from x to y.

Proof:

By definition.

Formula:

$\Psi_{D-Edge-Closed}(X) = \neg(\exists x \exists y (x \in X) \wedge (y \notin X) \wedge E(x, y))$

<u>Quasi-Path(X,x,y)</u>

Problem definition:

Given a directed graph G, two vertices x and y in G and a set X of edges, X, x and y satisfy Quasi-Path(X,x,y) iff

1. X is a subset of the edges of G
2. x and y are distinct
3. x is the origin of a unique edge in X and the target of none
4. y is the target of a unique edge in X and the origin of none
5. every vertex other than x or y that is incident to an edge of X is the origin of a unique edge in X and the target of a unique edge in X.

Verbal definition:

As above.

Proof:

By definition.

Formula:

$\Psi_{Quasi-Path}(X, x, y) =$

1. $\forall u \forall v [X(u, v) \Leftrightarrow E(u, v)] \wedge$

2. $(x \neq y) \wedge$

3. $\exists u[X(x,u) \wedge \forall v(X(x,v) \Rightarrow (v=u))] \wedge \neg \exists u X(u,x) \wedge$

4. $\exists u[X(u,y) \wedge \forall v(X(v,y) \Rightarrow (v=u))] \wedge \neg \exists u X(y,u) \wedge$

5. $\forall u[(u \neq x \wedge u \neq y \wedge \exists v(X(u,v) \vee X(v,u))) \Rightarrow \{\exists v[X(u,v) \wedge \forall w(X(u,w) \Rightarrow (v = w))] \wedge (v[X(v,u) \wedge \forall w(X(w,u) \Rightarrow (v=w))]\}$

Quasi-Path'(X,x,y)

Problem definition:

Given an undirected graph G, two vertices x and y in G and a set X of vertices, X, x and y satisfy Quasi-Path(X,x,y) iff x and y are distinct and are both in X, x and y are each adjacent to exactly one element of X and every vertex in $X\backslash\{x,y\}$ is adjacent to exactly two elements of X.

Verbal definition:

As above.

Proof:

By definition.

Formula: $(Quasi-Path'(X,x,y) = X(x) \wedge X(y) \wedge (x \neq y) \wedge \exists u[(X(x,u) \vee X(u,x)) \wedge \forall v((X(x,v) \vee X(v,x)) \Rightarrow (v=u))] \wedge \exists u[(X(y,u) \vee X(u,y)) \wedge \forall v((X(y,v) \vee X(v,y)) \Rightarrow (v = u))] \wedge \forall u[((u \neq x) \wedge (u \neq y) \wedge X(u)) \Rightarrow \exists v \exists w\{(X(v,u) \vee X(u,v)) \wedge (X(w,u) \vee X(u,w)) \wedge \forall s((X(s,u) \vee X(u,s)) \Rightarrow ((s=v) \vee (s=w))\}$

4 Proofs for the table entries:

In this section we give the proofs for our mapping of the graph properties that are listed in section 2 into their table entries. For each property, we give a *verbal* definition that is easy to write as a SOL formula, prove that it is equivalent to the problem's definition that was given in section 2, and analyze the "quantifier complexity" of the formula.
In some of the definitions we include other formulas that are defined in this section or in section 3 as sub-formulas.

Connectivity:

It is known:

Connectivity $\notin \exists X^1$ in GRAPH1.

We will show:

Connectivity $\in \exists X^1 \forall Y^1 \exists Z^1$ in GRAPH1.

Verbal definition:

The graph G satisfies Connectivity iff for every subset X of the vertices of G, X is either trivial (that is, equal to the set of vertices of G or to the empty set) or not edge-closed.

Proof:

$\Rightarrow$:

Suppose the graph G is connected and contains a non-trivial subset U of its vertices that is edge-closed. Let x be a vertex in U and y be a vertex not in U. (Such x and y exist because U is non-trivial). Since the graph is connected, there is a path between x and y. Let z be the vertex on this path that is closest to x and not in U, and let z' be the vertex that is right before z in the path (that is, adjacent to z and closer to x on the path). z and z' are connected, and therefore U is not edge-closed, in contradiction to our assumptions. Therefore, if G satisfies connectivity, every non-trivial subset of its set of vertices is not edge-closed.

$\Leftarrow$:

Suppose every non-trivial subset of the set of vertices of the graph G is not edge-closed. Given two vertices x and y, we will show that it is possible to construct a path between x and y:

1. Let U = $\{x\}$.

2. [U is not trivial (because U $\neq \emptyset$ and y $\notin$ U). Therefore, according to the assumption, U is not edge-closed.] Select an arch from x to a vertex v $\notin$ U, and add v to U.

3. Repeat step 2 as long as y $\notin$ U.

Note:

(1) According to the way v is selected in step 2, all of the vertices that are added to U are connected to x by a path.

(2) As long as y $\notin$ U, U is not trivial and therefore step 2 can be performed.

Conclusion: Since the set of vertices of the graph is finite, and by (2), y will be added to U at some point. Therefore, according to (1), x and y are connected by a path.

Formula:

$\Phi_{Connectivity} = \forall X(\forall x(x \in X) \vee \forall y(y \notin X) \vee \neg\Phi_{Edge-Closed}(X))$

Quantifier complexity:

$\Phi_{Edge-Closed}(X) \in FOL \Rightarrow \Phi_{Connectivity} \in \forall Y^1 \subseteq \exists X^1 \exists Y^1 \exists Z^1$ in GRAPH1.

Question:

is Connectivity $\in \exists X^2$?

Non-Connectivity:

We will show:

Non-Connectivity $\in \exists X^1$ in GRAPH1.

Verbal definition:

The graph G satisfies Non-Connectivity iff there exists a proper, non-empty subset X of the vertices of G such that X is edge-closed.

Proof:

This definition is the exact opposite of the definition we have supplied above for Connectivity, and since the proof for Connectivity is "iff", it proves this definition as well.

Formula:

$\Phi_{Non-Connectivity} = \exists X(\exists x(x \in X) \wedge \exists y(y \notin X) \wedge \Phi_{Edge-Closed}(X))$

Quantifier complexity:

$\Phi_{Edge-Closed}(X) \in FOL \Rightarrow \Phi_{Non-Connectivity} \in \exists X^1$ in GRAPH1.

R̲eachability(x,y):

We will show:

Reachability(x,y) $\in \exists X^1$ in GRAPH1.

Verbal definition:

The graph G satisfies Reachability(x,y) iff

(1) x = y OR:

(2) x $\neq$ y and there exists a subset X of the vertices of G such that Quasi-Path'(X,x,y) is satisfied.

Proof:

$\Rightarrow$:

Suppose the graph G satisfies Reachability(x,y).

Case 1: x = y. The then above holds.

Case 2: x $\neq$ y. Then there exists a simple path between x and y. Let U be the set of edges of this path. x and y are adjacent to exactly one edge of U. For every other vertex v in G, v is either in the path, in which case it is adjacent to exactly 2 edges of U (since the path is simple), or it is not in the path, in which case it is not adjacent to any edge of U. Therefore, Quasi-Path'(U,x,y) is satisfied. $\Leftarrow$: Suppose that x=y or: x $\neq$ y and the graph G satisfies Quasi-Path'(X,x,y). Case 1: x=y. Therefore, Reachability(x,y) is satisfied by the existence of an empty path connecting x and y. Case 2: x $\neq$ y and the graph G satisfies Quasi-Path'(X,x,y). Let Q be the set of edges that forms the quasi-path. x is adjacent to exactly one edge of Q. In addition, every other edge in the graph is adjacent to either none or exactly two of the edges of Q. Therefore, if the vertex v that is connected to x by an edge in Q is not y, it must be connected to another vertex (different from x) by an edge in Q. Similarly, any vertex ($\neq$ y) that is connected to x by a path constructed by edges that are in Q, must be connected to another vertex that is not already on this path. This path can only end by an edge that is adjacent to y, and therefore x and y are connected by a path, and hence Reachability(x,y) is satisfied.

Formula:

$\Phi_{Reachability}(x, y) = \exists X \Phi_{QuasiPath'}(X, x, y)$

Quantifier complexity::

$\Phi_{QuasiPath'}(X, x, y) \in FOL \Rightarrow \Phi_{Reachability} \in \exists X^1$ in GRAPH1.

Un-Reachability(x,y):

We will show:

$Un - Reachability(x, y) \in \exists X^1$ in G1.

Verbal definition:

The graph G satisfies Un-Reachability(x,y) iff there exists a subset X of the vertices of G such that x is in X, y is not in X and X is Edge-Closed.

Proof:

$\Rightarrow$:

Suppose the graph G satisfies Un-Reachability(x,y). Let $U = \{x\}$. Now, as long as it is still possible, select an edge connecting a vertex in U with a vertex w that is not in U and add w to U.

Note:

(1) At the end of this process, U is edge-closed.

(2) Since there is no path connecting x and y, y will not be added to U. Therefore, U satisfies the conditions required above.

$\Leftarrow$:

Suppose that there is a subset U of the vertices of G such that $x \in U, y \notin U$ and U is edge-closed. Therefore, there is no edge connecting a vertex from U with a vertex that is not in U, and hence there cannot be a path between x and y.

Formula:

$\Psi_{Un-Reachability}(x, y) = \exists X(\exists x(x \in X) \wedge \exists y(y \notin X) \wedge Edge - Closed(X))$

Quantifier complexity:

$\Psi_{Edge-Closed}(X) \in FOL \Rightarrow \Phi_{Un-Reachability}(x, y) \in \exists X^1$ in GRAPH1.

D-Reachability(x,y):

It is known:

D-Reachability(x,y) $\notin \exists X^1$ in GRAPH1. [2]

We will show:

D-Reachability(x,y) $\in \exists X^2$ in GRAPH1.

Verbal definition:

The graph G satisfies D-Reachability(x,y) iff

(1) x = y OR:

(2) x $\neq$ y and there exists a subset X of the vertices of G such that Quasi-Path(X,x,y) is satisfied.

Proof:

$\Rightarrow$:

Suppose the graph G satisfies Reachability(x,y).

Case 1: x = y. Then the above holds.

Case 2:

x $\neq$ y. Then there exists a simple directed path from x to y. Let U be the set of edges of this path. x is the source of exactly one edge in U, y is the target of exactly one edge in U. For every other vertex v in G, v is either in the path, in which case it is the source of exactly one edge in U and the target of exactly one edge in U (since the path is simple), or it is not in the path, in which case it is not adjacent to any edge of U. Therefore, Quasi-Path(U,x,y) is satisfied.

$\Leftarrow$:

Suppose that x=y or: x$\neq$ y and the graph G satisfies Quasi-Path(X,x,y).

Case 1:

x=y. Therefore, Reachability(x,y) is satisfied by the existence of the empty path from x to y.

Case 2:

x $\neq$ y and the graph G satisfies Quasi-Path(X,x,y). Let Q be the set of edges that form the quasi-path. x is the source of exactly one edge of Q. In addition, every other edge in the graph is either not adjacent to any edge of Q or it is the source of exactly one edge of Q and the target of exactly one edge of Q. Therefore, if the vertex v that is the target of the edge from Q that leaves x, it must be the source of another edge in Q, whose target is a vertex that is different from x. Similarly, any vertex ($\neq$ y) that is connected to x by a directed path that is directed away from x and constructed by edges that are in Q, must be connected to another vertex that is not already on this path. This path can only end by an edge whose target is y, and therefore there is a path from x to y, and hence D-Reachability(x,y) is satisfied.

Formula:

$\Psi_{D-Reachability}(x,y) = \exists X \Psi_{QuasiPath(X,x,y)}$

Quantifier complexity:

X is a set of pairs.

$\Psi_{QuasiPath}(X,x,y) \in FOL \Rightarrow \Psi_{D-Reachability}(x,y) \in \exists X^2$ in GRAPH1.

D-Un-Reachability(x,y):

We will show:

D-Un-Reachability(x,y) $\in \exists X^1$ in GRAPH1.

Verbal definition:

The graph G satisfies D-Un-Reachability(x,y) iff there exists a subset X of the vertices of G such that x is in X, y is not in X and X is D-Edge-Closed.

Proof:

$\Rightarrow$:

Suppose G satisfies D-Un-Reachability(x,y). Let $U = \{x\}$. Now, as long as it is still possible, select an edge directed from a vertex in U to a vertex w that is not in U and add w to U.

Note:

(1) At the end of this process, U is D-Edge-Closed.

(2) Since there is no path connecting x and y, y will not be added to U. Therefore, U satisfies the conditions required above.

$\Leftarrow$:

Suppose that there is a subset U of the vertices of G such that x $\in$ U, y $\notin$ U and U is D-Edge-Closed. Therefore, there is no edge directed from a vertex in U to a vertex that is not in U, and hence there cannot be a path from x to y.

Formula:

$\Psi_{D-Un-Reachability}(x,y) = \exists X(\exists x(x \in X) \wedge \exists y(y \notin X) \wedge \Psi_{D-Edge-Closed}(X))$

Quantifier complexity::

$D-Edge-Closed \in FOL \Rightarrow \Psi_{D-Reachability}(x,y) \in \exists X^1$ in GRAPH1.

k-Seperability(a,b):

We will show:

k-Seperability(x,y) $\in \exists X^1$ in GRAPH1.

Verbal definition:

Given a graph G and two vertices a,b the property of K-Seperability is satisfied iff there are two subsets X and Y of the set of vertices of G such that:

1. neither a nor b is in X$\cap$Y
2. X $\cup$ Y is exactly the set of vertices of G
3. $\mid$ X $\cap$ Y $\mid \leq$ k
4. There is no edge between a vertex u $\in$ X $\backslash\{X \cap Y\}$ and a vertex v $\in$ Y $\backslash\{X \cap Y\}$

Proof:

$\Rightarrow$:

Suppose the graph G satisfies k-Seperability(a,b) $\Rightarrow$ there is a set S of at most k vertices such that eliminating the vertices in S from the graph seperates a and b. Let X be the set containing S, a and all of the vertices that are accessible from a by a path that does not pass through a vertex of S and let Y be the set containing S and all of the vertices that are not in X.

By the construction of X and Y, conditions 1,2 and 3 hold.

Suppose condition 4 does not hold. Therefore, there is an edge between a vertex u $\in$ X $\backslash\{X \cap Y\}$ and a vertex v $\in$ Y $\backslash\{X \cap Y\}$. By the construction of X, u is accessible from a by a path that does not path through S. Therefore, v is also accessible from a by such a path and hence v $\in$ X, in contradiction to our assumption. Therefore, condition 4 holds as well.

$\Leftarrow$:

Suppose there exist two subsets of the set of vertices of the graph such that the four conditions above hold. Eliminate from the graph the set of vertices $X \cap Y$ (thus eliminating no more than k vertices). Suppose that in the remaining graph, a and b are connected. Then in the original graph there is an edge between a vertex from X $\backslash \{X \cap Y\}$ and a vertex from Y $\backslash \{X \cap Y\}$

Formula:

$\Psi_{K-Seperability}(x,y) = \exists X \exists Y\{$ 1. $[(x \in X) \vee (x \notin Y)] \wedge [(y \in Y) \vee (y \notin X)] \wedge$ 2. $[\forall z((z \in X) \vee (z \in Y))] \wedge$ 3. $[\forall v_1, \cdots, \forall v_{k+1}((v_1 \in X) \wedge \cdots \wedge (v_{k+1} \in X \wedge (v_1 \in Y) \wedge \cdots \wedge (v_{k+1} \in Y)) \Rightarrow ((v_1 = v_2) \vee \cdots \vee (v_1 = v_{k+1}) \vee (v_2 = v_3) \vee \cdots \vee (v_2 = v_{k+1}) \vee \cdots \vee (v_k = v_{k+1}))] \wedge$ 4. $[\forall a \forall b((a \in X \wedge (a \notin Y) \wedge (b \in Y) \wedge (b \notin X) \Rightarrow \neg E(a,b))]\}$

Quantifier complexity:

$\Psi_{k-Seperability}(x,y) \in \exists X^1$ in GRAPH1.

Non-k-Seperability(a,b):

We will show:

Non-k-Seperability(x,y) $\in \exists X^1$ in GRAPH1. *Verbal definition:*

The graph G satisfies Non-k-Seperability(a,b) iff: 1. a = b, OR 2. a and b are connected by an edge, OR 3. There are at least K+1 disjoint paths that connect a and b.

Proof:

$\Rightarrow$:

Supppose the graph G satisfies Non-k-Seperability(a,b).

Case 1: a = b. Then condition 1 holds.

Case 2: E(a,b). Then condition 2 holds.

Case 3: a $\neq$ b and $\neg$E(a,b) and there are more than k disjoint paths between a and b. Then condition 3 holds.

Case 4: a $\neq$ b and $\neg$E(a,b) and there are less than k+1 disjoint paths between a and b. In this case, there is a set of k (or less) vertices through which every path from a to b passes.

Therefore, Non-k-Seperability(a,b) is not satisfied, in contradiction to the assumption.

$\Leftarrow$:

Suppose one the three conditions above holds.

If a=b or E(a,b), then there is no set of vertices whose elimination can seperate a and b, and Non-k-Seperability(a,b) is satisfied.

If there are k+1 or more disjoint paths between a and b, then eliminating any set of vertices of size k or less that does not cointain a or b, will not cut off all of the paths between a and b, and hence Non-k-Seperability(a,b) is satisfied.

Formula:

$\Psi_{Non-k-Seperability}(x,y) = (x = y) \vee E(x,y) \vee \Psi_{(k+1)-Disjoint-paths}(x,y)$.

Quantifier complexity:

$\Psi_{(k+1)-Disjoint-path} \in \forall X^1 \exists X^1 \Rightarrow \Psi_{Non-k-Seperability} \in \forall X^1 \exists X^1$

Tree:

We will show:

Tree $\in \exists X^1 \forall Y^1 \exists Z^1$ in GRAPH1.

Verbal definition: The graph G satisfies Tree iff:

1. it does not contain a cycle of 3 vertices, and
2. it is connected, and
3. every two distinct vertices that are not adjacent are 1-seperable.

Proof:

Recall that one of the properties of a tree is that there is exactly one path between every two distinct vertices. (2) ensures, by definition, that there is at least one path between every two edges in the graph. (3) ensures, by definition, that there is at most one path between every two edges that are not adjacent. What is left to show is that there is no more than one path between two distinct edges that are adjacent. Suppose that vertices x and y are adjacent and that there are two or more paths between them. Therefore, there is a cycle containing x and y. Take the smallest such cycle. Since there are no parallel edges in the graph, this cycle is of size 3 or more. The first term in our formula excludes the possibility of a cycle of size 3. $\Rightarrow$ The cycle is of size 4 or more. Since this is the smallest cycle containing x and y, there are no chords in the cycle $\Rightarrow$ There exist two non-adjacent edges, u and v, in the cycle. $\Rightarrow$ u and v are distinct, non-adjacent and there are 2 distinct paths between them, in contradiction to (3).

Formula:

$\Psi_{Tree} = (\neg\Psi_{3-cycle}) \wedge \Psi_{Connectivity} \wedge [(\forall x \forall y((x \neq y) \wedge \neg E(x,y)) \Rightarrow \Psi_{1-seperability}(x,y)]$

Quantifier complexity: $3 - Cycle \in \exists X^1$

$Connectivity \in \exists X^1 \forall Y^1 \exists Z^1$

$\Psi_{1-Seperability} \in \exists X^1$

$\Rightarrow \Psi_{Tree} \in \exists X^1 \forall Y^1 \exists Z^1$

Non-Tree:

We will show:

Non-Tree $\in \exists X^1$ in GRAPH1.

Verbal definition:

The graph G satisfies Non-Tree iff:

1. it is not connected, or
2. it contains a cycle

Proof:

Recall that one of the basic properties of a tree is that it is connected and cycle-free. This formula asserts that the graph is either not connected or contains a cycle. Therefore, it must not be a tree.

Formula:

$\Psi_{Not-Tree} = \Psi_{Non-Connectivity} \vee \Psi_{Cycle}$

Quantifier complexity:

$\Psi_{Cycle}, \Psi_{2-Disjoint-paths} \in \exists X^1$

$Non - Connectivity \in \exists X^1$

$\Rightarrow \Psi_{Non-Tree} \in \exists X^1$

3-Colorability:

We will show:

3-Colorability $\in \exists X^1$ in GRAPH1. *Verbal definition:*

The graph G satisfies 3-colorability iff there exists a partition of the set of vertices of G into 3 subsets such that there are no two adjacent vertices in the same subset.

Proof:

$\Rightarrow$:

Suppose G satisfies 3-Colorability. Then there exists a coloring of the vertices of G by 3 colors such that no two adjacent vertices are colored by the same color. Let X be the set of vertices colored by the first color, Y the set of vertices colored by the second color and Z the set of vertices colored by the third color. X, Y and Z are disjoint, since every vertex is colored by exactly one color. In addition, there are no two adjacent vertices in the same group X, Y or Z, according to the properties of the coloring.

$\Leftarrow$:

Suppose there is a partition of the vertices of G into 3 subsets such that there are no two adjacent vertices in the same subset. Since this is a partition, every vertex is in exactly one such subset. Color the vertices of each subset by a different color. Since there are no two adjacent vertices in the same subset, this will be a legal 3-coloring of the vertices of G.

Formula:

$\Psi_{3-Colorability} = \exists X \exists Y \exists Z [\Psi_{3-Disjoint-Sets}(X,Y,Z) \wedge [\neg \exists x((x \notin X) \wedge (x \notin Y) \wedge (x \notin Z)] \wedge [\forall x((x \in X) \vee (x \in Y) \vee (x \in Z))] \wedge [\neg \exists x \exists y(E(x,y) \wedge (((x \in X) \wedge (y \in X)) \vee ((y \in Y) \wedge (y \in Y)) \vee ((x \in Y) \wedge (y \in Y))))]$

Quantifier complexity:

$\Psi_{3-Disjoint-Sets} \in$ FOL and the rest is $\in \exists X^1$. $\Rightarrow Psi_{3-Colorability} \in \exists X^1$

<u>**Non-3-Colorability:**</u>

It is known:

NP = coNP iff Non-3-Colorability is expressible by any formula in the bottom-most row of our table.

It is known:

$\Psi_{Non-3-Colorability} \notin \exists X^1$ in GRAPH1. [3] By Default, $\Psi_{Non-3-Colorability} = \neg\Psi_{Non-3-Colorability} \in \forall X^1 \exists X^1$.

Question:

Is $\Psi_{Non-3-Colorability} \in \forall X^2$?

Non-Chordality:

We will show:

$\Psi_{Non-Chordality} \in \exists X^1$ in GRAPH1.

Verbal definition:

The graph G satisfies Non-Chordality iff there exists a subset X of the vertices of G such that there are two distinct vertices x and y that are in X and are not connected by an edge, there exists a vertex z that is not in X and is connected by an edge to x and by another edge to y and not connected by an edge to any other vertex of X, and in addition, G satisfies Quasi-Path'(X,x,y).

Proof:

$\Rightarrow$:

Suppose G satisfies Non-Chordality. Then there exists an induced subgraph of G that is a cycle of length at least 4. Let z be one of the vertices of the smallest such cycle and let X be the set of all vertices of that cycle except for z. The cycle is of length at least 4 $\Rightarrow$ —X— $\geq$ 3. Since the sub-graph induced by X$\setminus\{z\}$ is a cycle of length at least 4 (and does not contain smaller cycles within it), there exist exactly two vertices that are adjacent to z. Let them be x and y. Since X is a cycle short of one vertex, there is a path between x and y in X, and therefore Quazi-Path'(X,x,y) is satisfied. Since the cycle is of length at least 4, the path between x and y is of length at least 2, so they are not adjacent.

$\Leftarrow$:

Suppose there is a subset X of the vertices of graph G as described above. Quasi-Path'(X,x,y) implies that there is a path between x and y within X. Since x and y are not adjacent, this path is of length at least 2. The fact that z is not in X and that it is adjacent to x and to y implies that the above mentioned path along with z and the edges connecting it to x and y form a cycle of length at least 4. Since z is not adjacent to any other vertex in X, there are no chords in this cycle and it is the smallest possible among this group of vertices.

Formula:

$\Psi_{Non-Chordality} = \exists X[\exists x \exists y \exists z(x \neq y \wedge x \in X \wedge y \in X \wedge z \notin X \wedge \neg E(x,y) \wedge E(x,z) \wedge E(z,y) \forall u\{(u \in X \wedge u \neq x \wedge u \neq y) \Rightarrow \neg E(u,z)\} \wedge Psi_{Quasi-Path'}(X,x,y)]$

Quantifier complexity:

$\Psi_{Non-Chordality} \in \exists X^1$ in GRAPH1.

Chordality:

Verbal definition:

A graph G satisfies Chordality iff the graph does not satisfy Non-Chordality.

Proof:

By definition. *Formula:*

$\Psi_{Chordality} = \Psi_{Non-Chordality}$

Quantifier complexity:

$\Psi_{Non-Chordality} \in \exists X^1 \Rightarrow sobyasimpletransformation, \Psi_{Chordality} \in \forall X^1 \subseteq \exists X^1 \forall Y^1 \exists Z^1$

k-Disjoint-Paths($x_1, x_2 \cdots x_k; y_1, y_2 \cdots y_k$):

Problem definition:

Given a graph G, the property of k-Disjoint-Paths($x_1, x_2 \cdots x_k; y_1, y_2 \cdots y_k$) is satisfied iff $x_1 \neq y_1, x_2 \neq y_2, \cdots, x_k \neq y_k$ and there exist k paths $p_1, \cdots, p_k$ linking x_1 to y_1, x_2 to y_2,$\cdots$, x_k to y_k respectively, such that every vertex common to two paths p_i and p_j, $j \neq i$, belongs to $\{x_i, y_i\} \cap \{x_j, y_j\}$.

Claim:

$\Psi_{K-Disjoint-paths}(x_1, \cdots x_k, y_1 \cdots y_k) = \exists X_1 \cdots \exists X_k[(\wedge_{i \neq j}(x \in X_i) \Rightarrow \neg(x \in X_j)) \wedge (\wedge_i \Psi_{Reachability}(X_i, x_i, y_i))$

Proof:

Say G_i is the sub-graph of G induced by X_i. If y_i is reachable from x_i in the sub-graph G_i mark the path on G. X_i are unique (doesn't have any joint vertex) so the paths marked on G will be disjoint.

Quantifier complexity::

The way this is written is just like k times Reachability which is $\exists X^1$.

Non-k-Disjoint-Paths($x_1, x_2 \cdots x_k; y_1, y_2 \cdots y_k$):

Problem definition:

Given a graph G, the property of Non-k-Disjoint-Paths($x_1, x_2 \cdots x_k; y_1, y_2 \cdots y_k$) is satisfied iff the property of k-Disjoint-Paths($x_1, x_2 \cdots x_k; y_1, y_2 \cdots y_k$) is not satisfied.

Quantifier complexity:

By default it is $\forall X^1 \exists X^1$ (by negating k-Disjoint-Paths and then transforming $\neg\exists \rightarrow \forall\neg$).

D-Connectivity:

Claim:

$\Psi_{D-Connectivity} = \forall X(\forall x(x \in X) \vee \forall y(y \notin X) \vee \neg\Psi_{D-Edge-Closed}(X))$

Correctness:

Say G satisfy $\Psi_{D-Connectivity}$. Then starting from any vertex in X, X can never be edge closed until it captures all G so we will just add one of the edges that go out from X and it's second vertex $\Rightarrow$ any vertex is directedly connected to any other. *Quantifier complexity:*

Obviously $\forall X^1$.

D-Non-Connectivity:

Problem definition:

Claim:

$\Psi_{D-Non-Connectivity} = \exists X(\exists x((x \in X)((y(y(X) \wedge \Psi_{D-Edge-Closed}(X))$

Correctness:

Just the negation of the former.

Quantifier complexity:

Obviously $\exists X^1$.

References: [1] Bruno Courcelle: On the expression of graph properties in some fragments of monadic second-order logic. [2] Ajtai, R. Fagin Reachability is harder for directed than for undirected finite graphs, J. of symbolic logic 55 (1990). [3] Cosmadakis: Logical reducability and monadic NP, Proceeding of 33rd IEEE symp. on foundations of computer science, 1993, pp.52-61.