FORMALIZED MATHEMATICS
Volume 14, Number 4, Pages 187–206
University of Białystok, 2006

Recognizing Chordal Graphs: Lex BFS and MCS[1]

Broderick Arneson
University of Alberta
Edmonton, Canada

Piotr Rudnicki
University of Alberta
Edmonton, Canada

Summary. We are formalizing the algorithm for recognizing chordal graphs by lexicographic breadth-first search as presented in [13, Section 3 of Chapter 4, pp. 81–84]. Then we follow with a formalization of another algorithm serving the same end but based on maximum cardinality search as presented by Tarjan and Yannakakis [25].

This work is a part of the MSc work of the first author under supervision of the second author. We would like to thank one of the anonymous reviewers for very useful suggestions.

MML identifier: `LEXBFS`, version: `7.8.03 4.75.958`

The notation and terminology used in this paper are introduced in the following articles: [28], [11], [26], [32], [33], [35], [30], [10], [7], [8], [20], [29], [4], [2], [14], [23], [12], [3], [6], [9], [18], [15], [19], [16], [17], [24], [21], [1], [5], [31], [27], [22], and [34].

1. Preliminaries

The following propositions are true:

(1) Let A, B be elements of $\mathbb{N}$, X be a non empty set, and F be a function from $\mathbb{N}$ into X. If F is one-to-one, then $\overline{\overline{\{F(w); w \text{ ranges over elements of } \mathbb{N}\colon A \leq w \ \wedge\ w \leq A+B\}}} = B+1$.

(2) For all natural numbers n, m, k such that $m \leq k$ and $n < m$ holds $k -' m < k -' n$.

[1]This work has been partially supported by the NSERC grant OGP 9207.

ISSN 1426–2630

(3) For all natural numbers n, k such that $n < k$ holds $(k -' (n+1)) + 1 = k -' n$.

(4) For all natural numbers n, m, k such that $k \neq 0$ holds $(n + m \cdot k) \div k = (n \div k) + m$.

Let S be a set. We say that S has finite elements if and only if:

(Def. 1) Every element of S is finite.

Let us note that there exists a set which is non empty and has finite elements and there exists a subset of $2^{\mathbb{N}}$ which is non empty and finite and has finite elements.

Let S be a set with finite elements. One can check that every element of S is finite.

Let f, g be functions. The functor $f[\cup]g$ yielding a function is defined by:

(Def. 2) $\mathrm{dom}(f[\cup]g) = \mathrm{dom}\, f \cup \mathrm{dom}\, g$ and for every set x such that $x \in \mathrm{dom}\, f \cup \mathrm{dom}\, g$ holds $(f[\cup]g)(x) = f(x) \cup g(x)$.

The following three propositions are true:

(5) For all natural numbers m, n, k holds $m \in \mathrm{Seg}\, k \setminus \mathrm{Seg}(k -' n)$ iff $k -' n < m$ and $m \leq k$.

(6) For all natural numbers n, k, m such that $n \leq m$ holds $\mathrm{Seg}\, k \setminus \mathrm{Seg}(k -' n) \subseteq \mathrm{Seg}\, k \setminus \mathrm{Seg}(k -' m)$.

(7) For all natural numbers n, k such that $n < k$ holds $(\mathrm{Seg}\, k \setminus \mathrm{Seg}(k -' n)) \cup \{k -' n\} = \mathrm{Seg}\, k \setminus \mathrm{Seg}(k -' (n+1))$.

Let f be a binary relation. We say that f is natsubset yielding if and only if:

(Def. 3) $\mathrm{rng}\, f \subseteq 2^{\mathbb{N}}$.

Let us mention that there exists a function which is finite-yielding and natsubset yielding.

Let f be a finite-yielding natsubset yielding function and let x be a set. Then $f(x)$ is a finite subset of $\mathbb{N}$.

One can prove the following proposition

(8) For every ordinal number X and for all finite subsets a, b of X such that $a \neq b$ holds $(a, 1)\text{-bag} \neq (b, 1)\text{-bag}$.

Let F be a natural-yielding function, let S be a set, and let k be a natural number. The functor $F.\mathrm{incSubset}(S, k)$ yielding a natural-yielding function is defined by the conditions (Def. 4).

(Def. 4)(i) $\mathrm{dom}(F.\mathrm{incSubset}(S, k)) = \mathrm{dom}\, F$, and

(ii) for every set y holds if $y \in S$ and $y \in \mathrm{dom}\, F$, then $(F.\mathrm{incSubset}(S, k))(y) = F(y) + k$ and if $y \notin S$, then $(F.\mathrm{incSubset}(S, k))(y) = F(y)$.

Let n be an ordinal number, let T be a connected term order of n, and let B be a non empty finite subset of Bags n. The functor $\max(B,T)$ yields a bag of n and is defined as follows:

(Def. 5) $\max(B,T) \in B$ and for every bag x of n such that $x \in B$ holds $x \leq_T \max(B,T)$.

Let O be an ordinal number. Observe that InvLexOrder O is connected.

2. Miscellany on Graphs

Let G be a graph. Note that there exists a vertex sequence of G which is non empty and one-to-one.

Let G be a graph and let V be a non empty vertex sequence of G. A walk of G is called a walk of V if:

(Def. 6) It.vertexSeq() $= V$.

Let G be a graph and let V be a non empty one-to-one vertex sequence of G. One can check that every walk of V is path-like.

We now state two propositions:

(9) For every graph G and for all walks W_1, W_2 of G such that W_1 is trivial and W_1.last() $= W_2$.first() holds W_1.append(W_2) $= W_2$.

(10) Let G, H be graphs, A, B, C be sets, G_1 be a subgraph of G induced by A, H_1 be a subgraph of H induced by B, G_2 be a subgraph of G_1 induced by C, and H_2 be a subgraph of H_1 induced by C. Suppose $G =_G H$ and $A \subseteq B$ and $C \subseteq A$ and C is a non empty subset of the vertices of G. Then $G_2 =_G H_2$.

Let G be a v-graph. We say that G is natural v-labeled if and only if:

(Def. 7) The vlabel of G is natural-yielding.

3. Graphs with Two Vertex Labels

The natural number V2-LabelSelector is defined by:

(Def. 8) V2-LabelSelector $= 8$.

Let G be a graph structure. We say that G is v2-labeled if and only if:

(Def. 9) V2-LabelSelector $\in \operatorname{dom} G$ and there exists a function f such that G(V2-LabelSelector) $= f$ and $\operatorname{dom} f \subseteq$ the vertices of G.

Let us note that there exists a graph structure which is graph-like, weighted, elabeled, vlabeled, and v2-labeled.

A v2-graph is a v2-labeled graph. A vv-graph is a vlabeled v2-labeled graph.

Let G be a v2-graph. The v2-label of G yields a function and is defined as follows:

(Def. 10) The v2-label of $G = G$(V2-LabelSelector).

Next we state the proposition

(11) For every v2-graph G holds dom (the v2-label of G) $\subseteq$ the vertices of G.

Let G be a graph and let X be a set. Note that G.set(V2-LabelSelector, X) is graph-like.

We now state the proposition

(12) For every graph G and for every set X holds G.set(V2-LabelSelector, X) $=_G G$.

Let G be a finite graph and let X be a set.

Note that G.set(V2-LabelSelector, X) is finite.

Let G be a loopless graph and let X be a set.

Observe that G.set(V2-LabelSelector, X) is loopless.

Let G be a trivial graph and let X be a set.

Note that G.set(V2-LabelSelector, X) is trivial.

Let G be a non trivial graph and let X be a set. One can check that G.set(V2-LabelSelector, X) is non trivial.

Let G be a non-multi graph and let X be a set. One can check that G.set(V2-LabelSelector, X) is non-multi.

Let G be a non-directed-multi graph and let X be a set. One can verify that G.set(V2-LabelSelector, X) is non-directed-multi.

Let G be a connected graph and let X be a set.

Note that G.set(V2-LabelSelector, X) is connected.

Let G be an acyclic graph and let X be a set.

One can verify that G.set(V2-LabelSelector, X) is acyclic.

Let G be a v-graph and let X be a set.

One can check that G.set(V2-LabelSelector, X) is vlabeled.

Let G be a e-graph and let X be a set. Observe that G.set(V2-LabelSelector, X) is elabeled.

Let G be a w-graph and let X be a set. Observe that G.set(V2-LabelSelector, X) is weighted.

Let G be a v2-graph and let X be a set.

One can verify that G.set(VLabelSelector, X) is v2-labeled.

Let G be a graph, let Y be a set, and let X be a partial function from the vertices of G to Y. Observe that G.set(V2-LabelSelector, X) is v2-labeled.

Let G be a graph and let X be a many sorted set indexed by the vertices of G. Observe that G.set(V2-LabelSelector, X) is v2-labeled.

Let G be a graph. One can verify that G.set(V2-LabelSelector, $\emptyset$) is v2-labeled.

Let G be a v2-graph. We say that G is natural v2-labeled if and only if:

(Def. 11) The v2-label of G is natural-yielding.

We say that G is finite v2-labeled if and only if:

(Def. 12) The v2-label of G is finite-yielding.

We say that G is natsubset v2-labeled if and only if:

(Def. 13) The v2-label of G is natsubset yielding.

One can check that there exists a weighted elabeled vlabeled v2-labeled graph which is finite, natural v-labeled, finite v2-labeled, natsubset v2-labeled, and chordal and there exists a weighted elabeled vlabeled v2-labeled graph which is finite, natural v-labeled, natural v2-labeled, and chordal.

Let G be a natural v-labeled v-graph. Observe that the vlabel of G is natural-yielding.

Let G be a natural v2-labeled v2-graph. Observe that the v2-label of G is natural-yielding.

Let G be a finite v2-labeled v2-graph. Observe that the v2-label of G is finite-yielding.

Let G be a natsubset v2-labeled v2-graph. One can verify that the v2-label of G is natsubset yielding.

Let G be a vv-graph and let v, x be sets. One can check that G.labelVertex(v, x) is v2-labeled.

Next we state the proposition

(13) For every vv-graph G and for all sets v, x holds the v2-label of $G =$ the v2-label of G.labelVertex(v, x).

Let G be a natural v-labeled vv-graph, let v be a set, and let x be a natural number. Observe that G.labelVertex(v, x) is natural v-labeled.

Let G be a natural v2-labeled vv-graph, let v be a set, and let x be a natural number. Observe that G.labelVertex(v, x) is natural v2-labeled.

Let G be a finite v2-labeled vv-graph, let v be a set, and let x be a natural number. Note that G.labelVertex(v, x) is finite v2-labeled.

Let G be a natsubset v2-labeled vv-graph, let v be a set, and let x be a natural number. One can check that G.labelVertex(v, x) is natsubset v2-labeled.

Let G be a graph. Note that there exists a subgraph of G which is vlabeled and v2-labeled.

Let G be a v2-graph and let G_2 be a v2-labeled subgraph of G. We say that G_2 inherits v2-label if and only if:

(Def. 14) The v2-label of $G_2 =$ (the v2-label of G)$\restriction$(the vertices of G_2).

Let G be a v2-graph. Note that there exists a v2-labeled subgraph of G which inherits v2-label.

Let G be a v2-graph. A v2-subgraph of G is a v2-labeled subgraph of G inheriting v2-label.

Let G be a vv-graph. Note that there exists a vlabeled v2-labeled subgraph of G which inherits vlabel and v2-label.

Let G be a vv-graph. A vv-subgraph of G is a vlabeled v2-labeled subgraph of G inheriting vlabel and v2-label.

Let G be a natural v-labeled v-graph. Note that every v-subgraph of G is natural v-labeled.

Let G be a graph and let V, E be sets. Observe that there exists a subgraph of G induced by V and E which is weighted, elabeled, vlabeled, and v2-labeled.

Let G be a vv-graph and let V, E be sets. Observe that there exists a vlabeled v2-labeled subgraph of G induced by V and E which inherits vlabel and v2-label.

Let G be a vv-graph and let V, E be sets. A (V, E)-induced vv-subgraph of G is a vlabeled v2-labeled subgraph of G induced by V and E inheriting vlabel and v2-label.

Let G be a vv-graph and let V be a set. A V-induced vv-subgraph of G is a $(V, G.\mathrm{edgesBetween}(V))$-induced vv-subgraph of G.

4. More on Graph Sequences

Let s be a many sorted set indexed by $\mathbb{N}$. We say that s is iterative if and only if:

(Def. 15) For all natural numbers k, n such that $s(k) = s(n)$ holds $s(k+1) = s(n+1)$.

Let G_3 be a many sorted set indexed by $\mathbb{N}$. We say that G_3 is eventually constant if and only if:

(Def. 16) There exists a natural number n such that for every natural number m such that $n \leq m$ holds $G_3(n) = G_3(m)$.

Let us observe that there exists a many sorted set indexed by $\mathbb{N}$ which is halting, iterative, and eventually constant.

The following proposition is true

(14) For every many sorted set G_4 indexed by $\mathbb{N}$ such that G_4 is halting and iterative holds G_4 is eventually constant.

One can check that every many sorted set indexed by $\mathbb{N}$ which is halting and iterative is also eventually constant.

The following proposition is true

(15) For every many sorted set G_4 indexed by $\mathbb{N}$ such that G_4 is eventually constant holds G_4 is halting.

Let us mention that every many sorted set indexed by $\mathbb{N}$ which is eventually constant is also halting.

One can prove the following two propositions:

(16) Let G_4 be an iterative eventually constant many sorted set indexed by $\mathbb{N}$ and n be a natural number. If $G_4.\mathrm{Lifespan}() \leq n$, then $G_4(G_4.\mathrm{Lifespan}()) = G_4(n)$.

(17) Let G_4 be an iterative eventually constant many sorted set indexed by $\mathbb{N}$ and n, m be natural numbers. If G_4.Lifespan() $\leq n$ and $n \leq m$, then $G_4(m) = G_4(n)$.

Let G_3 be a v-graph sequence. We say that G_3 is natural v-labeled if and only if:

(Def. 17) For every natural number x holds $G_3(x)$ is natural v-labeled.

Let G_3 be a graph sequence. We say that G_3 is chordal if and only if:

(Def. 18) For every natural number x holds $G_3(x)$ is chordal.

We say that G_3 has fixed vertices if and only if:

(Def. 19) For all natural numbers n, m holds the vertices of $G_3(n)$ = the vertices of $G_3(m)$.

We say that G_3 is v2-labeled if and only if:

(Def. 20) For every natural number x holds $G_3(x)$ is v2-labeled.

Let us observe that there exists a graph sequence which is weighted, elabeled, vlabeled, and v2-labeled.

A v2-graph sequence is a v2-labeled graph sequence. A vv-graph sequence is a vlabeled v2-labeled graph sequence.

Let G_5 be a v2-graph sequence and let x be a natural number. Note that $G_5(x)$ is v2-labeled.

Let G_5 be a v2-graph sequence. We say that G_5 is natural v2-labeled if and only if:

(Def. 21) For every natural number x holds $G_5(x)$ is natural v2-labeled.

We say that G_5 is finite v2-labeled if and only if:

(Def. 22) For every natural number x holds $G_5(x)$ is finite v2-labeled.

We say that G_5 is natsubset v2-labeled if and only if:

(Def. 23) For every natural number x holds $G_5(x)$ is natsubset v2-labeled.

Let us mention that there exists a weighted elabeled vlabeled v2-labeled graph sequence which is finite, natural v-labeled, finite v2-labeled, natsubset v2-labeled, and chordal and there exists a weighted elabeled vlabeled v2-labeled graph sequence which is finite, natural v-labeled, natural v2-labeled, and chordal.

Let G_4 be a v-graph sequence and let x be a natural number. Then $G_4(x)$ is a v-graph.

Let G_5 be a natural v-labeled v-graph sequence and let x be a natural number. Observe that $G_5(x)$ is natural v-labeled.

Let G_5 be a natural v2-labeled v2-graph sequence and let x be a natural number. One can check that $G_5(x)$ is natural v2-labeled.

Let G_5 be a finite v2-labeled v2-graph sequence and let x be a natural number. One can verify that $G_5(x)$ is finite v2-labeled.

Let G_5 be a natsubset v2-labeled v2-graph sequence and let x be a natural number. Note that $G_5(x)$ is natsubset v2-labeled.

Let G_5 be a chordal graph sequence and let x be a natural number. One can check that $G_5(x)$ is chordal.

Let G_4 be a v-graph sequence and let n be a natural number. Then $G_4(n)$ is a v-graph.

Let G_4 be a finite v-graph sequence and let n be a natural number. One can check that $G_4(n)$ is finite.

Let G_4 be a vv-graph sequence and let n be a natural number. Then $G_4(n)$ is a vv-graph.

Let G_4 be a finite vv-graph sequence and let n be a natural number. One can verify that $G_4(n)$ is finite.

Let G_4 be a chordal vv-graph sequence and let n be a natural number. Note that $G_4(n)$ is chordal.

Let G_4 be a natural v-labeled vv-graph sequence and let n be a natural number. One can check that $G_4(n)$ is natural v-labeled.

Let G_4 be a finite v2-labeled vv-graph sequence and let n be a natural number. Note that $G_4(n)$ is finite v2-labeled.

Let G_4 be a natsubset v2-labeled vv-graph sequence and let n be a natural number. One can check that $G_4(n)$ is natsubset v2-labeled.

Let G_4 be a natural v2-labeled vv-graph sequence and let n be a natural number. Observe that $G_4(n)$ is natural v2-labeled.

5. Vertices Numbering Sequences

Let G_3 be a v-graph sequence. We say that G_3 has initially empty v-label if and only if:

(Def. 24) The vlabel of $G_3(0) = \emptyset$.

We say that G_3 is adding one at a step if and only if the condition (Def. 25) is satisfied.

(Def. 25) Let n be a natural number. Suppose $n < G_3$.Lifespan(). Then there exists a set w such that $w \notin \text{dom}\,(\text{the vlabel of } G_3(n))$ and the vlabel of $G_3(n+1) = (\text{the vlabel of } G_3(n)) {+\cdot} (w \dot{\longmapsto} (G_3.\text{Lifespan}() -' n))$.

Let G_3 be a v-graph sequence. We say that G_3 is v-label numbering if and only if the condition (Def. 26) is satisfied.

(Def. 26) G_3 is iterative, eventually constant, finite, natural v-labeled, and adding one at a step and has fixed vertices and initially empty v-label.

One can check that there exists a v-graph sequence which is iterative, eventually constant, finite, natural v-labeled, and adding one at a step and has fixed vertices and initially empty v-label.

Let us observe that there exists a v-graph sequence which is v-label numbering.

One can check the following observations:

* every v-graph sequence which is v-label numbering is also iterative,
* every v-graph sequence which is v-label numbering is also eventually constant,
* every v-graph sequence which is v-label numbering is also finite,
* every v-graph sequence which is v-label numbering has also fixed vertices,
* every v-graph sequence which is v-label numbering is also natural v-labeled,
* every v-graph sequence which is v-label numbering has also initially empty v-label, and
* every v-graph sequence which is v-label numbering is also adding one at a step.

A v-label numbering sequence is a v-label numbering v-graph sequence.

Let G_3 be a v-label numbering sequence and let n be a natural number. The functor G_3 .PickedAt n yields a set and is defined by:

(Def. 27)(i) G_3 .PickedAt n = choose(the vertices of $G_3(0)$) if $n \geq G_3$.Lifespan(),

(ii) G_3 .PickedAt $n \notin$ dom (the vlabel of $G_3(n)$) and the vlabel of $G_3(n+1)$ = (the vlabel of $G_3(n)$)$+\cdot$((G_3 .PickedAt n)$\dot{\longmapsto}$(G_3.Lifespan() $-'$ n)), otherwise.

The following propositions are true:

(18) Let G_3 be a v-label numbering sequence and n be a natural number. If $n < G_3$.Lifespan(), then G_3 .PickedAt $n \in G_3(n+1)$.labeledV() and $G_3(n+1)$.labeledV() $= G_3(n)$.labeledV() $\cup$ {G_3 .PickedAt n}.

(19) Let G_3 be a v-label numbering sequence and n be a natural number. If $n < G_3$.Lifespan(), then (the vlabel of $G_3(n+1)$)(G_3 .PickedAt n) = G_3.Lifespan() $-'$ n.

(20) For every v-label numbering sequence G_3 and for every natural number n such that $n \leq G_3$.Lifespan() holds card($G_3(n)$.labeledV()) $= n$.

(21) For every v-label numbering sequence G_3 and for every natural number n holds rng (the vlabel of $G_3(n)$) = Seg(G_3.Lifespan())$\setminus$Seg(G_3.Lifespan()$-'$ n).

(22) Let G_3 be a v-label numbering sequence, n be a natural number, and x be a set. Then (the vlabel of $G_3(n)$)(x) $\leq G_3$.Lifespan() and if $x \in G_3(n)$.labeledV(), then $1 \leq$ (the vlabel of $G_3(n)$)(x).

(23) Let G_3 be a v-label numbering sequence and n, m be natural numbers. Suppose G_3.Lifespan() $-'$ $n < m$ and $m \leq G_3$.Lifespan(). Then there exists a vertex v of $G_3(n)$ such that $v \in G_3(n)$.labeledV() and (the vlabel

of $G_3(n))(v) = m$.

(24) Let G_3 be a v-label numbering sequence and m, n be natural numbers. If $m \leq n$, then the vlabel of $G_3(m) \subseteq$ the vlabel of $G_3(n)$.

(25) For every v-label numbering sequence G_3 and for every natural number n holds the vlabel of $G_3(n)$ is one-to-one.

(26) Let G_3 be a v-label numbering sequence, m, n be natural numbers, and v be a set. Suppose $v \in G_3(m)$.labeledV() and $v \in G_3(n)$.labeledV(). Then (the vlabel of $G_3(m))(v) =$ (the vlabel of $G_3(n))(v)$.

(27) Let G_3 be a v-label numbering sequence, v be a set, and m, n be natural numbers. If $v \in G_3(m)$.labeledV() and (the vlabel of $G_3(m))(v) = n$, then G_3 .PickedAt(G_3.Lifespan() $-'$ n) $= v$.

(28) Let G_3 be a v-label numbering sequence and m, n be natural numbers. If $n < G_3$.Lifespan() and $n < m$, then G_3 .PickedAt $n \in G_3(m)$.labeledV() and (the vlabel of $G_3(m))(G_3$ .PickedAt $n) = G_3$.Lifespan() $-'$ n.

(29) Let G_3 be a v-label numbering sequence, m be a natural number, and v be a set. Suppose $v \in G_3(m)$.labeledV(). Then G_3.Lifespan() $-'$ (the vlabel of $G_3(m))(v) < m$ and G_3.Lifespan() $-'$ $m <$ (the vlabel of $G_3(m))(v)$.

(30) Let G_3 be a v-label numbering sequence, i be a natural number, and a, b be sets. Suppose $a \in G_3(i)$.labeledV() and $b \in G_3(i)$.labeledV() and (the vlabel of $G_3(i))(a) <$ (the vlabel of $G_3(i))(b)$. Then $b \in$ $G_3(G_3$.Lifespan() $-'$ (the vlabel of $G_3(i))(a))$.labeledV().

(31) Let G_3 be a v-label numbering sequence, i be a natural number, and a, b be sets. Suppose $a \in G_3(i)$.labeledV() and $b \in G_3(i)$.labeledV() and (the vlabel of $G_3(i))(a) <$ (the vlabel of $G_3(i))(b)$. Then $a \notin$ $G_3(G_3$.Lifespan() $-'$ (the vlabel of $G_3(i))(b))$.labeledV().

6. Lexicographical Breadth-First Search

Let G be a graph. The functor LexBFS:Init G yields a natural v-labeled finite v2-labeled natsubset v2-labeled vv-graph and is defined as follows:

(Def. 28) LexBFS:Init G $=$ G.set(VLabelSelector, $\emptyset$).set(V2-LabelSelector, (the vertices of $G) \longmapsto \emptyset)$.

Let G be a finite graph. Then LexBFS:Init G is a finite natural v-labeled finite v2-labeled natsubset v2-labeled vv-graph.

Let G be a finite finite v2-labeled natsubset v2-labeled vv-graph. Let us assume that dom (the v2-label of G) $=$ the vertices of G. The functor LexBFS:PickUnnumbered G yields a vertex of G and is defined by:

(Def. 29)(i) LexBFS:PickUnnumbered G = choose(the vertices of G) if dom (the vlabel of G) = the vertices of G,

(ii) there exists a non empty finite subset S of $2^{\mathbb{N}}$ and there exists a non empty finite subset B of Bags $\mathbb{N}$ and there exists a function F such that $S =$ rng F and $F =$ (the v2-label of G)$\restriction$((the vertices of G) $\setminus$ dom (the vlabel of G)) and for every finite subset x of $\mathbb{N}$ such that $x \in S$ holds $(x, 1)$-bag $\in B$ and for every set x such that $x \in B$ there exists a finite subset y of $\mathbb{N}$ such that $y \in S$ and $x = (y, 1)$-bag and LexBFS:PickUnnumbered $G =$ choose$(F^{-1}(\{\text{support}\max(B, \text{InvLexOrder}\,\mathbb{N})\}))$, otherwise.

Let G be a vv-graph, let v be a set, and let k be a natural number. The functor LexBFS:LabelAdjacent(G, v, k) yielding a vv-graph is defined as follows:

(Def. 30) LexBFS:LabelAdjacent$(G, v, k) = G$.set(V2-LabelSelector, (the v2-label of G)$[\cup]$((G.adjacentSet($\{v\}$)) $\setminus$ dom (the vlabel of G) $\longmapsto \{k\}$)).

Next we state four propositions:

(32) Let G be a vv-graph, v, x be sets, and k be a natural number. If $x \notin G$.adjacentSet($\{v\}$), then (the v2-label of G)(x) = (the v2-label of LexBFS:LabelAdjacent(G, v, k))(x).

(33) Let G be a vv-graph, v, x be sets, and k be a natural number. Suppose $x \in$ dom (the vlabel of G). Then (the v2-label of G)(x) = (the v2-label of LexBFS:LabelAdjacent(G, v, k))(x).

(34) Let G be a vv-graph, v, x be sets, and k be a natural number. Suppose $x \in G$.adjacentSet($\{v\}$) and $x \notin$ dom (the vlabel of G). Then (the v2-label of LexBFS:LabelAdjacent(G, v, k))(x) = (the v2-label of G)$(x) \cup \{k\}$.

(35) Let G be a vv-graph, v be a set, and k be a natural number. Suppose dom (the v2-label of G) = the vertices of G. Then dom (the v2-label of LexBFS:LabelAdjacent(G, v, k)) = the vertices of G.

Let G be a finite natural v-labeled finite v2-labeled natsubset v2-labeled vv-graph, let v be a vertex of G, and let k be a natural number. Then LexBFS:LabelAdjacent(G, v, k) is a finite natural v-labeled finite v2-labeled natsubset v2-labeled vv-graph.

Let G be a finite natural v-labeled finite v2-labeled natsubset v2-labeled vv-graph, let v be a vertex of G, and let n be a natural number. The functor LexBFS:Update(G, v, n) yielding a finite natural v-labeled finite v2-labeled natsubset v2-labeled vv-graph is defined by:

(Def. 31) LexBFS:Update$(G, v, n) =$
LexBFS:LabelAdjacent$(G.\text{labelVertex}(v, G.\text{order}() -' n), v, G.\text{order}() -' n)$.

Let G be a finite natural v-labeled finite v2-labeled natsubset v2-labeled vv-graph. The functor LexBFS:Step G yields a finite natural v-labeled finite v2-labeled natsubset v2-labeled vv-graph and is defined as follows:

(Def. 32) $$\text{LexBFS:Step}\,G = \begin{cases} G, & \text{if } G.\text{order}() \leq \text{card}\,\text{dom}\,(\text{the vlabel of } G), \\ \text{LexBFS:Update}(G, \text{LexBFS:PickUnnumbered}\,G, & \\ \quad \text{card}\,\text{dom}\,(\text{the vlabel of } G)), & \text{otherwise.} \end{cases}$$

Let G be a finite graph. The functor LexBFS:CSeq G yields a finite natural v-labeled finite v2-labeled natsubset v2-labeled vv-graph sequence and is defined by:

(Def. 33) (LexBFS:CSeq G)(0) = LexBFS:Init G and for every natural number n holds (LexBFS:CSeq G)($n+1$) = LexBFS:Step(LexBFS:CSeq G)(n).

We now state the proposition

(36) For every finite graph G holds LexBFS:CSeq G is iterative.

Let G be a finite graph. Observe that LexBFS:CSeq G is iterative.

Next we state a number of propositions:

(37) For every graph G holds the vlabel of LexBFS:Init $G = \emptyset$.

(38) Let G be a graph and v be a set. Then dom (the v2-label of LexBFS:Init G) = the vertices of G and (the v2-label of LexBFS:Init G)(v) $= \emptyset$.

(39) For every graph G holds $G =_G$ LexBFS:Init G.

(40) Let G be a finite finite v2-labeled natsubset v2-labeled vv-graph and x be a set. Suppose that

(i) $x \notin$ dom (the vlabel of G),

(ii) dom (the v2-label of G) = the vertices of G, and

(iii) dom (the vlabel of G) $\neq$ the vertices of G.

Then ((the v2-label of G)(x), 1) -bag $\leq_{\mathrm{InvLexOrder}\,\mathbb{N}}$ ((the v2-label of G)(LexBFS:PickUnnumbered G), 1) -bag .

(41) Let G be a finite finite v2-labeled natsubset v2-labeled vv-graph. Suppose dom (the v2-label of G) = the vertices of G and dom (the vlabel of G) $\neq$ the vertices of G. Then LexBFS:PickUnnumbered $G \notin$ dom (the vlabel of G).

(42) For every finite graph G and for every natural number n holds (LexBFS:CSeq G)(n) $=_G G$.

(43) For every finite graph G and for all natural numbers m, n holds (LexBFS:CSeq G)(m) $=_G$ (LexBFS:CSeq G)(n).

(44) Let G be a finite graph and n be a natural number. Suppose card dom (the vlabel of (LexBFS:CSeq G)(n)) $<$ G.order(). Then the vlabel of (LexBFS:CSeq G)($n+1$) = (the vlabel of (LexBFS:CSeq G)(n))+·(LexBFS:PickUnnumbered(LexBFS:CSeq G)(n) $\dot{\longmapsto}$(G.order() $-'$ card dom (the vlabel of (LexBFS:CSeq G)(n)))).

(45) For every finite graph G and for every natural number n holds dom (the v2-label of (LexBFS:CSeq G)(n)) = the vertices of (LexBFS:CSeq G)(n).

(46) For every finite graph G and for every natural number n such that $n \leq$ G.order() holds card dom (the vlabel of (LexBFS:CSeq G)(n)) $= n$.

(47) For every finite graph G and for every natural number n such that G.order() $\leq$ n holds (LexBFS:CSeq G)(G.order()) =

(LexBFS:CSeq G)(n).

(48) For every finite graph G and for all natural numbers m, n such that G.order() $\leq m$ and $m \leq n$ holds (LexBFS:CSeq G)(m) = (LexBFS:CSeq G)(n).

(49) For every finite graph G holds LexBFS:CSeq G is eventually constant.

Let G be a finite graph. Note that LexBFS:CSeq G is eventually constant.

We now state two propositions:

(50) Let G be a finite graph and n be a natural number. Then dom (the vlabel of (LexBFS:CSeq G)(n)) = the vertices of (LexBFS:CSeq G)(n) if and only if G.order() $\leq n$.

(51) For every finite graph G holds (LexBFS:CSeq G).Lifespan() = G.order().

Let G be a finite chordal graph and let i be a natural number. One can check that (LexBFS:CSeq G)(i) is chordal.

Let G be a finite chordal graph. One can check that LexBFS:CSeq G is chordal.

One can prove the following proposition

(52) For every finite graph G holds LexBFS:CSeq G is v-label numbering.

Let G be a finite graph. Note that LexBFS:CSeq G is v-label numbering.

We now state several propositions:

(53) For every finite graph G and for every natural number n such that $n <$ G.order() holds LexBFS:CSeq G .PickedAt n = LexBFS:PickUnnumbered(LexBFS:CSeq G)(n).

(54) Let G be a finite graph and n be a natural number. Suppose $n <$ G.order(). Then there exists a vertex w of (LexBFS:CSeq G)(n) such that

(i) w = LexBFS:PickUnnumbered(LexBFS:CSeq G)(n), and

(ii) for every set v holds if $v \in G$.adjacentSet($\{w\}$) and $v \notin$ dom (the vlabel of (LexBFS:CSeq G)(n)), then (the v2-label of (LexBFS:CSeq G)($n+1$))(v) = (the v2-label of (LexBFS:CSeq G)(n))(v) $\cup$ {G.order() $-'$ n} and if $v \notin G$.adjacentSet($\{w\}$) or $v \in$ dom (the vlabel of (LexBFS:CSeq G)(n)), then (the v2-label of (LexBFS:CSeq G)($n+1$))(v) = (the v2-label of (LexBFS:CSeq G)(n))(v).

(55) Let G be a finite graph, i be a natural number, and v be a set. Then (the v2-label of (LexBFS:CSeq G)(i))(v) $\subseteq$ Seg(G.order()) $\setminus$ Seg(G.order() $-'$ i).

(56) Let G be a finite graph, x be a set, and i, j be natural numbers. If $i \leq j$, then (the v2-label of (LexBFS:CSeq G)(i))(x) $\subseteq$ (the v2-label of (LexBFS:CSeq G)(j))(x).

(57) Let G be a finite graph, m, n be natural numbers, and x, y be sets. Suppose $n <$ G.order() and $n < m$ and y = LexBFS:PickUnnumbered(LexBFS:CSeq G)(n) and $x \notin$ dom (the vlabel of

(LexBFS:CSeq G)(n)) and $x \in G$.adjacentSet($\{y\}$). Then G.order() $-'$ $n \in$ (the v2-label of (LexBFS:CSeq G)(m))(x).

(58) Let G be a finite graph and m, n be natural numbers. Suppose $m < n$. Let x be a set. Suppose G.order() $-'$ $m \notin$ (the v2-label of (LexBFS:CSeq G)($m + 1$))(x). Then G.order() $-'$ $m \notin$ (the v2-label of (LexBFS:CSeq G)(n))(x).

(59) Let G be a finite graph and m, n, k be natural numbers. Suppose $k < n$ and $n \leq m$. Let x be a set. Suppose G.order() $-'$ $k \notin$ (the v2-label of (LexBFS:CSeq G)(n))(x). Then G.order() $-'$ $k \notin$ (the v2-label of (LexBFS:CSeq G)(m))(x).

(60) Let G be a finite graph, m, n be natural numbers, and x be a vertex of (LexBFS:CSeq G)(m). Suppose $n \in$ (the v2-label of (LexBFS:CSeq G)(m))(x). Then there exists a vertex y of (LexBFS:CSeq G)(m) such that LexBFS:PickUnnumbered(LexBFS:CSeq G) (G.order() $-'$ n) $= y$ and $y \notin$ dom (the vlabel of (LexBFS:CSeq G)(G.order() $-'$ n)) and $x \in G$.adjacentSet($\{y\}$).

Let G_4 be a finite natural v-labeled vv-graph sequence. Then G_4.Result() is a finite natural v-labeled vv-graph.

The following four propositions are true:

(61) For every finite graph G holds (LexBFS:CSeq G).Result().labeledV() $=$ the vertices of G.

(62) For every finite graph G holds (the vlabel of (LexBFS:CSeq G).Result())$^{-1}$ is a vertex scheme of G.

(63) Let G be a finite graph, i, j be natural numbers, and a, b be vertices of (LexBFS:CSeq G)(i). Suppose that

(i) $a \in$ dom (the vlabel of (LexBFS:CSeq G)(i)),

(ii) $b \in$ dom (the vlabel of (LexBFS:CSeq G)(i)),

(iii) (the vlabel of (LexBFS:CSeq G)(i))(a) $<$ (the vlabel of (LexBFS:CSeq G)(i))(b), and

(iv) $j = G$.order() $-'$ (the vlabel of (LexBFS:CSeq G)(i))(b).

Then ((the v2-label of (LexBFS:CSeq G)(j))(a), 1)-bag $\leq_{\mathrm{InvLexOrder}\,\mathbb{N}}$ ((the v2-label of (LexBFS:CSeq G)(j))(b), 1)-bag.

(64) Let G be a finite graph, i, j be natural numbers, and v be a vertex of (LexBFS:CSeq G)(i). Suppose $j \in$ (the v2-label of (LexBFS:CSeq G)(i))(v). Then there exists a vertex w of (LexBFS:CSeq G)(i) such that $w \in$ dom (the vlabel of (LexBFS:CSeq G)(i)) and (the vlabel of (LexBFS:CSeq G)(i))(w) $= j$ and $v \in G$.adjacentSet($\{w\}$).

Let G be a natural v-labeled v-graph. We say that G has property *L3* if and only if the condition (Def. 34) is satisfied.

(Def. 34) Let a, b, c be vertices of G. Suppose that $a \in \mathrm{dom}$ (the vlabel of G) and $b \in \mathrm{dom}$ (the vlabel of G) and $c \in \mathrm{dom}$ (the vlabel of G) and (the vlabel of G)(a) $<$ (the vlabel of G)(b) and (the vlabel of G)(b) $<$ (the vlabel of G)(c) and a and c are adjacent and b and c are not adjacent. Then there exists a vertex d of G such that

(i) $d \in \mathrm{dom}$ (the vlabel of G),

(ii) (the vlabel of G)(c) $<$ (the vlabel of G)(d),

(iii) b and d are adjacent,

(iv) a and d are not adjacent, and

(v) for every vertex e of G such that $e \neq d$ and e and b are adjacent and e and a are not adjacent holds (the vlabel of G)(e) $<$ (the vlabel of G)(d).

One can prove the following three propositions:

(65) For every finite graph G and for every natural number n holds (LexBFS:CSeq G)(n) has property *L3*.

(66) Let G be a finite chordal natural v-labeled v-graph. Suppose G has property *L3* and dom (the vlabel of G) = the vertices of G. Let V be a vertex scheme of G. If V^{-1} = the vlabel of G, then V is perfect.

(67) For every finite chordal vv-graph G holds (the vlabel of (LexBFS:CSeq G).Result())$^{-1}$ is a perfect vertex scheme of G.

7. The Maximum Cardinality Search Algorithm

Let G be a finite graph. The functor MCS:Init G yields a finite natural v-labeled natural v2-labeled vv-graph and is defined by:

(Def. 35) MCS:Init G = G.set(VLabelSelector, $\emptyset$).set(V2-LabelSelector, (the vertices of G) $\longmapsto 0$).

Let G be a finite natural v2-labeled vv-graph. Let us assume that dom (the v2-label of G) = the vertices of G. The functor MCS:PickUnnumbered G yields a vertex of G and is defined by:

(Def. 36)(i) MCS:PickUnnumbered G = choose(the vertices of G) if dom (the vlabel of G) = the vertices of G,

(ii) there exists a finite non empty natural-membered set S and there exists a function F such that $S = \mathrm{rng}\, F$ and F = (the v2-label of G)$\restriction$((the vertices of G) $\setminus$ dom (the vlabel of G)) and MCS:PickUnnumbered G = choose($F^{-1}(\{\max S\})$), otherwise.

Let G be a finite natural v2-labeled vv-graph and let v be a set. The functor MCS:LabelAdjacent(G, v) yields a finite natural v2-labeled vv-graph and is defined by:

(Def. 37) MCS:LabelAdjacent(G, v) = G.set(V2-LabelSelector, (the v2-label of G) .incSubset((G.adjacentSet($\{v\}$)) $\setminus$ dom (the vlabel of G), 1)).

Let G be a finite natural v-labeled natural v2-labeled vv-graph and let v be a vertex of G. Then MCS:LabelAdjacent(G, v) is a finite natural v-labeled natural v2-labeled vv-graph.

Let G be a finite natural v-labeled natural v2-labeled vv-graph, let v be a vertex of G, and let n be a natural number. The functor MCS:Update(G, v, n) yielding a finite natural v-labeled natural v2-labeled vv-graph is defined as follows:

(Def. 38) MCS:Update(G, v, n) = MCS:LabelAdjacent(G.labelVertex$(v, G$.order$() -' n), v)$.

Let G be a finite natural v-labeled natural v2-labeled vv-graph. The functor MCS:Step G yielding a finite natural v-labeled natural v2-labeled vv-graph is defined by:

(Def. 39) MCS:Step G = $\begin{cases} G, \text{ if } G.\text{order}() \leq \text{card dom (the vlabel of } G), \\ \text{MCS:Update}(G, \text{MCS:PickUnnumbered } G, \text{card dom} \\ \quad \text{(the vlabel of } G)), \text{ otherwise.} \end{cases}$

Let G be a finite graph. The functor MCS:CSeq G yields a finite natural v-labeled natural v2-labeled vv-graph sequence and is defined by:

(Def. 40) (MCS:CSeq G)(0) = MCS:Init G and for every natural number n holds (MCS:CSeq G)$(n + 1)$ = MCS:Step(MCS:CSeq G)(n).

The following proposition is true

(68) For every finite graph G holds MCS:CSeq G is iterative.

Let G be a finite graph. Observe that MCS:CSeq G is iterative.

We now state a number of propositions:

(69) For every finite graph G holds the vlabel of MCS:Init $G = \emptyset$.

(70) Let G be a finite graph and v be a set. Then dom (the v2-label of MCS:Init G) = the vertices of G and (the v2-label of MCS:Init G)$(v) = 0$.

(71) For every finite graph G holds $G =_G$ MCS:Init G.

(72) Let G be a finite natural v2-labeled vv-graph and x be a set. Suppose that
  - (i) $x \notin$ dom (the vlabel of G),
  - (ii) dom (the v2-label of G) = the vertices of G, and
  - (iii) dom (the vlabel of G) $\neq$ the vertices of G.

  Then (the v2-label of G)$(x) \leq$ (the v2-label of G)(MCS:PickUnnumbered G).

(73) Let G be a finite natural v2-labeled vv-graph. Suppose dom (the v2-label of G) = the vertices of G and dom (the vlabel of G) $\neq$ the vertices of G. Then MCS:PickUnnumbered $G \notin$ dom (the vlabel of G).

(74) Let G be a finite natural v2-labeled vv-graph and v, x be sets. If $x \notin G$.adjacentSet($\{v\}$), then (the v2-label of G)(x) = (the v2-label of

MCS:LabelAdjacent$(G, v))(x)$.

(75) Let G be a finite natural v2-labeled vv-graph and v, x be sets. Suppose $x \in$ dom (the vlabel of G). Then (the v2-label of $G)(x) =$ (the v2-label of MCS:LabelAdjacent$(G, v))(x)$.

(76) Let G be a finite natural v2-labeled vv-graph and v, x be sets. Suppose $x \in$ dom (the v2-label of G) and $x \in G$.adjacentSet$(\{v\})$ and $x \notin$ dom (the vlabel of G). Then (the v2-label of MCS:LabelAdjacent$(G, v))(x) =$ (the v2-label of $G)(x) + 1$.

(77) Let G be a finite natural v2-labeled vv-graph and v be a set. Suppose dom (the v2-label of G) = the vertices of G. Then dom (the v2-label of MCS:LabelAdjacent(G, v)) = the vertices of G.

(78) For every finite graph G and for every natural number n holds (MCS:CSeq $G)(n) =_G G$.

(79) For every finite graph G and for all natural numbers m, n holds (MCS:CSeq $G)(m) =_G$ (MCS:CSeq $G)(n)$.

Let G be a finite chordal graph and let n be a natural number. Observe that (MCS:CSeq $G)(n)$ is chordal.

Let G be a finite chordal graph. Observe that MCS:CSeq G is chordal.

One can prove the following propositions:

(80) For every finite graph G and for every natural number n holds dom (the v2-label of (MCS:CSeq $G)(n)$) = the vertices of (MCS:CSeq $G)(n)$.

(81) Let G be a finite graph and n be a natural number. Suppose card dom (the vlabel of (MCS:CSeq $G)(n)$) $< G$.order(). Then the vlabel of (MCS:CSeq $G)(n+1) =$ (the vlabel of (MCS:CSeq $G)(n)$) $+\cdot$(MCS:PickUnnumbered(MCS:CSeq $G)(n) \dot{\longmapsto} (G$.order()$-'$card dom (the vlabel of (MCS:CSeq $G)(n)$))).

(82) For every finite graph G and for every natural number n such that $n \leq G$.order() holds card dom (the vlabel of (MCS:CSeq $G)(n)$) $= n$.

(83) For every finite graph G and for every natural number n such that G.order() $\leq n$ holds (MCS:CSeq $G)(G$.order()$) =$ (MCS:CSeq $G)(n)$.

(84) For every finite graph G and for all natural numbers m, n such that G.order() $\leq m$ and $m \leq n$ holds (MCS:CSeq $G)(m) =$ (MCS:CSeq $G)(n)$.

(85) For every finite graph G holds MCS:CSeq G is eventually constant.

Let G be a finite graph. Observe that MCS:CSeq G is eventually constant.

The following propositions are true:

(86) Let G be a finite graph and n be a natural number. Then dom (the vlabel of (MCS:CSeq $G)(n)$) = the vertices of (MCS:CSeq $G)(n)$ if and only if G.order() $\leq n$.

(87) For every finite graph G holds (MCS:CSeq G).Lifespan() $= G$.order().

(88) For every finite graph G holds MCS:CSeq G is v-label numbering.

Let G be a finite graph. Note that MCS:CSeq G is v-label numbering.

Next we state three propositions:

(89) For every finite graph G and for every natural number n such that $n <$ G.order() holds MCS:CSeq G .PickedAt $n =$ MCS:PickUnnumbered(MCS:CSeq G)(n).

(90) Let G be a finite graph and n be a natural number. Suppose $n <$ G.order(). Then there exists a vertex w of (MCS:CSeq G)(n) such that

(i) $w =$ MCS:PickUnnumbered(MCS:CSeq G)(n), and

(ii) for every set v holds if $v \in G$.adjacentSet($\{w\}$) and $v \notin$ dom (the vlabel of (MCS:CSeq G)(n)), then (the v2-label of (MCS:CSeq G)($n+1$))(v) $=$ (the v2-label of (MCS:CSeq G)(n))(v) $+ 1$ and if $v \notin G$.adjacentSet($\{w\}$) or $v \in$ dom (the vlabel of (MCS:CSeq G)(n)), then (the v2-label of (MCS:CSeq G)($n+1$))(v) $=$ (the v2-label of (MCS:CSeq G)(n))(v).

(91) Let G be a finite graph, n be a natural number, and x be a set. Suppose $x \notin$ dom (the vlabel of (MCS:CSeq G)(n)). Then (the v2-label of (MCS:CSeq G)(n))(x) $=$ card((G.adjacentSet($\{x\}$)) $\cap$ dom (the vlabel of (MCS:CSeq G)(n))).

Let G be a natural v-labeled v-graph. We say that G has property T if and only if the condition (Def. 41) is satisfied.

(Def. 41) Let a, b, c be vertices of G. Suppose that $a \in$ dom (the vlabel of G) and $b \in$ dom (the vlabel of G) and $c \in$ dom (the vlabel of G) and (the vlabel of G)(a) $<$ (the vlabel of G)(b) and (the vlabel of G)(b) $<$ (the vlabel of G)(c) and a and c are adjacent and b and c are not adjacent. Then there exists a vertex d of G such that

(i) $d \in$ dom (the vlabel of G),

(ii) (the vlabel of G)(b) $<$ (the vlabel of G)(d),

(iii) b and d are adjacent, and

(iv) a and d are not adjacent.

We now state three propositions:

(92) For every finite graph G and for every natural number n holds (MCS:CSeq G)(n) has property T.

(93) For every finite graph G holds (LexBFS:CSeq G).Result() has property T.

(94) Let G be a finite chordal natural v-labeled v-graph. Suppose G has property T and dom (the vlabel of G) $=$ the vertices of G. Let V be a vertex scheme of G. If $V^{-1} =$ the vlabel of G, then V is perfect.

References

[1] Broderick Arneson and Piotr Rudnicki. Chordal graphs. *Formalized Mathematics*, 14(**3**):79–92, 2006.
[2] Grzegorz Bancerek. Cardinal numbers. *Formalized Mathematics*, 1(**2**):377–382, 1990.
[3] Grzegorz Bancerek. The fundamental properties of natural numbers. *Formalized Mathematics*, 1(**1**):41–46, 1990.
[4] Grzegorz Bancerek. The ordinal numbers. *Formalized Mathematics*, 1(**1**):91–96, 1990.
[5] Grzegorz Bancerek and Krzysztof Hryniewiecki. Segments of natural numbers and finite sequences. *Formalized Mathematics*, 1(**1**):107–114, 1990.
[6] Czesław Byliński. A classical first order language. *Formalized Mathematics*, 1(**4**):669–676, 1990.
[7] Czesław Byliński. Functions and their basic properties. *Formalized Mathematics*, 1(**1**):55–65, 1990.
[8] Czesław Byliński. Functions from a set to a set. *Formalized Mathematics*, 1(**1**):153–164, 1990.
[9] Czesław Byliński. The modification of a function by a function and the iteration of the composition of a function. *Formalized Mathematics*, 1(**3**):521–527, 1990.
[10] Czesław Byliński. Partial functions. *Formalized Mathematics*, 1(**2**):357–367, 1990.
[11] Czesław Byliński. Some basic properties of sets. *Formalized Mathematics*, 1(**1**):47–53, 1990.
[12] Agata Darmochwał. Finite sets. *Formalized Mathematics*, 1(**1**):165–167, 1990.
[13] M. Ch. Golumbic. *Algorithmic Graph Theory and Perfect Graphs.* Academic Press, New York, 1980.
[14] Jarosław Kotowicz. Monotone real sequences. Subsequences. *Formalized Mathematics*, 1(**3**):471–475, 1990.
[15] Gilbert Lee. Walks in Graphs. *Formalized Mathematics*, 13(**2**):253–269, 2005.
[16] Gilbert Lee. Weighted and Labeled Graphs. *Formalized Mathematics*, 13(**2**):279–293, 2005.
[17] Gilbert Lee and Piotr Rudnicki. On ordering of bags. *Formalized Mathematics*, 10(**1**):39–46, 2002.
[18] Gilbert Lee and Piotr Rudnicki. Alternative graph structures. *Formalized Mathematics*, 13(**2**):235–252, 2005.
[19] Yatsuka Nakamura, Piotr Rudnicki, Andrzej Trybulec, and Pauline N. Kawamoto. Preliminaries to circuits, I. *Formalized Mathematics*, 5(**2**):167–172, 1996.
[20] Takaya Nishiyama and Yasuho Mizuhara. Binary arithmetics. *Formalized Mathematics*, 4(**1**):83–86, 1993.
[21] Piotr Rudnicki. Little Bezout theorem (factor theorem). *Formalized Mathematics*, 12(**1**):49–58, 2004.
[22] Piotr Rudnicki and Andrzej Trybulec. Abian's fixed point theorem. *Formalized Mathematics*, 6(**3**):335–338, 1997.
[23] Piotr Rudnicki and Andrzej Trybulec. Multivariate polynomials with arbitrary number of variables. *Formalized Mathematics*, 9(**1**):95–110, 2001.
[24] Christoph Schwarzweller. Term orders. *Formalized Mathematics*, 11(**1**):105–111, 2003.
[25] R. E. Tarjan and M. Yannakakis. Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs. *SIAM J. Comput.*, 13(3):566–579, 1984.
[26] Andrzej Trybulec. Subsets of complex numbers. *To appear in Formalized Mathematics.*
[27] Andrzej Trybulec. Domains and their Cartesian products. *Formalized Mathematics*, 1(**1**):115–122, 1990.
[28] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[29] Andrzej Trybulec. Many-sorted sets. *Formalized Mathematics*, 4(**1**):15–22, 1993.
[30] Andrzej Trybulec. On the sets inhabited by numbers. *Formalized Mathematics*, 11(**4**):341–347, 2003.
[31] Wojciech A. Trybulec. Pigeon hole principle. *Formalized Mathematics*, 1(**3**):575–579, 1990.
[32] Zinaida Trybulec. Properties of subsets. *Formalized Mathematics*, 1(**1**):67–71, 1990.
[33] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.

[34] Edmund Woronowicz. Relations defined on sets. *Formalized Mathematics*, 1(**1**):181–186, 1990.

[35] Edmund Woronowicz and Anna Zalewska. Properties of binary relations. *Formalized Mathematics*, 1(**1**):85–89, 1990.

Received November 17, 2006