Dragan Mašulović

Introduction to Discrete Mathematics

Linz, October 2010

Chapter 1

Words and Sets

This chapter confronts us with the most basic abstract structures:

- words (or strings), which represent the simplest *ordered* structures, and
- sets (or collections), which represent the simplest *unordered* structures.

As we shall see, a *permutation* is nothing but a word over an appropriately chosen alphabet, while a *combination* is just a subset of a finite set. It is natural to ask why should one invent so complicated names for such simple objects. The answer is simple. In the dark past of Discrete Mathematics the terminology used to be as obscure as the ages that gave birth to it. Since the introduction of the names such as *permutation* and *combination* mathematics has gone a long way and brought many simplifications, both in terminology and understanding of the phenomena.

Throughout the course we shall use the following notation

$$\begin{aligned}
&\mathbb{N} = \{1,2,3,\ldots\} \quad \text{for the set of positive integers,}\\
&\mathbb{N}_0 = \{0,1,2,3,\ldots\} \quad \text{for the set of nonnegative integers, and}\\
&\mathbb{N}_0^\infty = \{0,1,2,3,\ldots\} \cup \{\infty\}.
\end{aligned}$$

The set $\mathbb{N}_0^\infty$ is a usual extension of $\mathbb{N}_0$ with the greatest element ∞: $x+\infty = \infty + x = x \cdot \infty = \infty \cdot x = \infty$ for all $x \in \mathbb{N}_0^\infty$, and $x < \infty$ for all $x \in \mathbb{N}_0$. Also, we define the *factorial* of an integer $n \in \mathbb{N}_0$ as usual:

$$\begin{aligned}
&0! = 1\\
&n! = 1 \cdot 2 \cdot \ldots \cdot n, \text{ for } n \geqslant 1.
\end{aligned}$$

1.1 Words

An *alphabet* is any finite nonempty set. Elements of an alphabet A will be referred to as *letters*, and a *word in* A is a string of symbols from A. More precisely, a *word of length* k *over an alphabet* A is any tuple from A^k. We follow a simple convention to omit commas and parentheses when writing words.

Example 1.1 Here are some words over an alphabet $A = \{a, b, n\}$: *banana*, *abba*, *aa*, or simply *n*. The first of the words has six letters, then comes a four-letter word, a two-letter word and finaly a word with only one letter.

We also alow words with no letters. On any alphabet there is precisely one such word called the *empty word* and denoted by ε. It is a word with length 0. It is important to note that words we deal with in this course are *formal words*, that is, strings of symbols to which no meaning is attached. So, from this point of view *nbbaaa* is just as good a word as *banana*. We shall leave the meaning of words to other branches of science and treat words just as plain and simple strings of letters.

Let w be a word over an alphabet A. The length of w will be denoted by $|w|$. For a letter $a \in A$, by $|w|_a$ we denote the number of occurences of a in w.

Example 1.2 Let $A = \{a, b, c, n\}$ and let $w =$ *banana* be a word over A. Then $|w| = 6$, $|w|_a = 3$, $|w|_b = 1$, $|w|_c = 0$ and $|w|_n = 2$.

There is not much structural theory behind such simple objects as words. The most exciting thing we can do at the moment is to try to count them.

Problem 1.3 *Let $A = \{a_1, a_2, \ldots, a_n\}$ be an alphabet with $n \geqslant 1$ letters and let $k \in \mathbb{N}_0$ be arbitrary.*

(a) How many words with k letters over A are there?

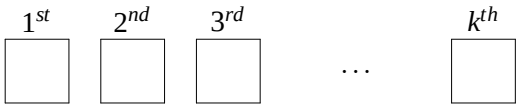
(b) How many words with k letters over A have the property that all the letters in the word are distinct?

(c) How many words over A have the property that every letter from A appears precisely once in the word?

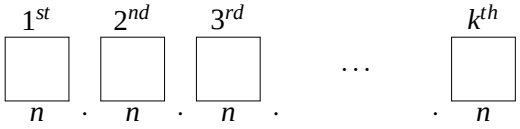
Solution. (a) The set of all words of length k over A is just A^k. Therefore, there are precisely $|A^k| = \underbrace{|A| \cdot \ldots \cdot |A|}_{k} = n^k$ such words. This is an instance of an important combinatorial principle:

The Product Principle: *If $A_1, \ldots, A_n$ are finite sets, then*
$$|A_1 \times \ldots \times A_n| = |A_1| \cdot \ldots \cdot |A_n|.$$

There is a less formal, but more useful way to see this. A word with k letters looks like this:

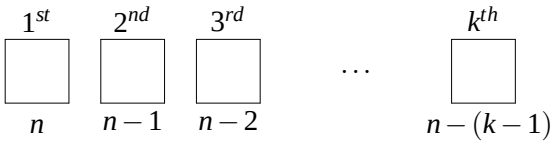


There are n candidates for the first position, n candidates for the second position, ..., n candidates for the kth position:



Alltogether, there are $\underbrace{n \cdot n \cdot \ldots \cdot n}_{k} = n^k$ possibilites.

(b) Let us again take the informal point of view. Firstly, there are n candidates for the first position, but only $n-1$ candidates for the second position, since the letter used on the first position is not allowed to appear on the second position. Then, there are $n-2$ candidates for the third position since the two letters used on the first two positions are a no-no, and so on. Finally, there will be $n-(k-1)$ candidates for the last position:



and putting it all together we get $n \cdot (n-1) \cdot \ldots \cdot (n-k+1) = \dfrac{n!}{(n-k)!}$ possibilites. Of course, this reasoning is valid as long as $k \leqslant n$. If $k > n$ no such word exists.

(c) If every letter from A is required to appear precisely once in the word, then the length of the word is n and all the letters have to be distinct. This is a special case of (b) where $k = n$ and there are $n!$ such words. □

Words where letters are not allowed to repeat are called *permutations of sets*. Words where letters can appear more than once constitute another kind of permutations — permutations of multisets — and we shall consider them in a separate section.

Definition 1.4 A *permutation of a set A* is a word over A where every letter from the alphabet appears precisely once in the word. A *k-permutation of a set A*, where $k \leqslant |A|$, is a word over A of length k where each letter from the alphabet is allowed to appear at most once (and therefore, all the letters in the word are distinct).

We shall now apply counting techniques discussed above to determine the number of all the subsets of a finite set. For a set A let $\mathscr{P}(A)$ denote the *power-set of A*, that is, the set of all the subsets of A:

$$\mathscr{P}(A) = \{X : X \subseteq A\}.$$

Let $|A| = n$ and $A = \{a_1, \ldots, a_n\}$. Then every subset B of A can be represented by a string $\chi(B)$ of 0's and 1's as follows:

$$\chi(B) = p_1 \ldots p_n, \quad \text{where} \quad p_i = \begin{cases} 0, & a_i \notin B, \\ 1, & a_i \in B. \end{cases}$$

The word $\chi(B)$ is called the *characteristic vector* of B. Words over the two-element alphabet $\{0,1\}$ will be particularly useful in the sequel. So, we shall refer to them as *01-words*.

Example 1.5 Let $A = \{a,b,c,d,e,f\}$ and $B = \{b,d,e\}$. Then $\chi(B) = 010110$ since $a \notin B$, $b \in B$, $c \notin B$ etc. Clearly, $\chi(\varnothing) = 000000$ and $\chi(A) = 111111$:

	a	b	c	d	e	f
$\varnothing$	0	0	0	0	0	0
B	0	1	0	1	1	0
A	1	1	1	1	1	1

Theorem 1.6 *Let A be a finite set with n elements. Then $|\mathscr{P}(A)| = 2^n$.*

Proof. The mapping $\chi : \mathscr{P}(A) \to \{0,1\}^n$ that takes a subset of A onto its characteristic vector is a bijection, so $|\mathscr{P}(A)|$ and $|\{0,1\}^n|$ have the same number of elements. We shall use this obvious but important fact on many occasions in the course:

> **The Bijection Principle:** *Whenever there is a bijection between two sets, they have the same number of elements.*

Therefore, $|\mathscr{P}(A)|$ equals the number of all words over $\{0,1\}$ whose length is n, so $|\mathscr{P}(A)| = 2^n$. □

The following principle is a sort of a negation of the Bijection Principle:

> **The Pigeon-Hole Principle:** *Suppose that A and B are nonempty finite sets such that $|A| > |B|$. Then there is no injective mapping from A into B.*

The name comes from a simple observation concerning pigeons and holes: if $n+1$ pigeons hide in n holes, then there is at least one hole with at least two pigeons in it.

Theorem 1.7 (Erdős, Szekeres 1935) *Every sequence of $k = mn+1$ distinct real numbers has an increasing subsequence of length $m+1$ or a decreasing subsequence of length $n+1$.*

Proof. Let $a_1, \ldots, a_k$ be a sequence of $k = mn+1$ distinct real numbers and assume that it has neither an increasing subsequence of length $m+1$ nor a decreasing subsequence of length $n+1$. Then every increasing subsequence of $a_1, \ldots, a_k$ is of length $\leqslant m$ and, similarly, every decreasing subsequence of $a_1, \ldots, a_k$ is of length $\leqslant n$.

For each i let l_i^+ denote the length of the longest increasing subsequence of $a_1, \ldots, a_k$ that starts with a_i and let l_i^- denote the length of the longest decreasing subsequence of $a_1, \ldots, a_k$ that starts with a_i. This establishes a mapping $f : \{1,\ldots,k\} \to \{1,\ldots,m\} \times \{1,\ldots,n\} : i \mapsto (l_i^+, l_i^-)$. Let us show that f is injective. Take any pair of indices $i \neq j$. Then $a_i \neq a_j$. If $a_i < a_j$ then $l_i^+ > l_j^+$ so $f(i) = (l_i^+, l_i^-) \neq (l_j^+, l_j^-) = f(j)$. Similarly, if $a_i > a_j$ then $l_i^- > l_j^-$ and we again conclude $f(i) \neq f(j)$. This shows that $i \neq j$ implies $f(i) \neq f(j)$ and thus f is an injective map from a k-element set into and mn-element set. But $k > mn$ and hence by the Pigeon-Hole Principle no such injective map can exist. Contradiction. □

1.2 Sets

For historical reasons, a k-element subset of an n-element set is called a *k-combination* of a set. The number of k-combinations of an n-element set is denoted by

$$\binom{n}{k} \qquad \text{[read: “}n\text{ choose }k\text{”].}$$

The pronounciation comes from the fact that this is the number of ways to choose k objects from a pool of n identical objects. If we let

$$\mathscr{P}_k(A) = \{B \in \mathscr{P}(A) : |B| = k\}$$

be the set of all k-subsets of A and if $|A| = n$, then, clearly,

$$\binom{n}{k} = |\mathscr{P}_k(A)|.$$

Theorem 1.8 *Let $n,k \geqslant 0$. If $n \geqslant k$ then $\binom{n}{k} = \dfrac{n!}{k!(n-k)!}$. Otherwise, $\binom{n}{k} = 0$.*

Proof. Let $n \geqslant k$ and let $A = \{1,2,\ldots,n\}$. Although sets seem to be simpler than words due to the lack of structure, ordered structures (words in this case) are always easier to count. Let $\mathscr{W}_k(A)$ be the set of all k-permutations of A and let $f : \mathscr{W}_k(A) \to \mathscr{P}_k(A)$ be the mapping defined by

$$f(a_1a_2\ldots a_k) = \{a_1,a_2,\ldots,a_k\}.$$

Since f maps $k!$ different words from $\mathscr{W}_k(A)$ onto the same element of $\mathscr{P}_k(A)$, e.g.

$$\left.\begin{array}{cccc} abcd & bacd & cabd & dabc \\ abdc & badc & cadb & dacb \\ acbd & bcad & cbad & dbac \\ acdb & bcda & cbda & dbca \\ adbc & bdac & cdab & dcab \\ adcb & bdca & cdba & dcba \end{array}\right\} \mapsto \{a,b,c,d\},$$

we easily conclude that

$$|\mathscr{P}_k(A)| = \frac{1}{k!} \cdot |\mathscr{W}_k(A)|.$$

We know that the number of k-permutations of an n-element set is $\frac{n!}{(n-k)!}$, so we finaly obtain that

$$\binom{n}{k} = |\mathscr{P}_k(A)| = \frac{1}{k!} \cdot \frac{n!}{(n-k)!}.$$

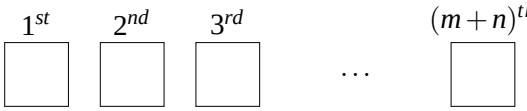
On the other hand, if $k > n$ then trivially $\binom{n}{k} = 0$ since an n-element set cannot have a subset with more than n elements. □

Problem 1.9 *How many 01-words of length $m+n$ are there if they are required to have precisely m zeros and precisely n ones?*

Solution. Consider a set $A = \{a_1,a_2,\ldots,a_{m+n}\}$ with $m+n$ elements. Then each 01-word of length $m+n$ with m zeros and n ones corresponds to an n-element subset of A. Therefore, the number of such 01-words equals the number of n-element subsets of A, which is

$$\binom{m+n}{n}.$$

Here is the other way to see this. Consider a string of $m+n$ empty boxes which are to be filled by m zeros and n ones:



We can choose m boxes in which to write zeros in $\binom{m+n}{m}$ ways. Then the remaining n boxes have to be filled by ones. □

Theorem 1.10 (a) $\binom{n}{k} = \binom{n}{n-k}$ *for all* $n \geqslant k \geqslant 0$*;*

(b) $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$ *for all* $n \geqslant k \geqslant 1$ *(Pascal's identity).*

Proof. (a) This follows by an easy calculation:

$$\binom{n}{n-k} = \frac{n!}{(n-k)!(n-(n-k))!} = \frac{n!}{(n-k)!k!} = \binom{n}{k}.$$

Such proofs are usually called *algebraic proofs*.

Most combinatorial identities can be proven in another way: we find an appropriate collection of objects and then count the elements of the collection in two different ways. The resulting expressions have to be equal because the collection is the same. Such proofs are usually called *combinatorial proofs*. The principle behind theis approach is called Double Counting:

Double Counting: *If the same set is counted in two different ways, the answers are the same.*

Let us provide a combinatorial proof of the same identity. Consider 01-words of length n with precisely k zeros. There are $\binom{n}{k}$ ways to choose k places out of n in which to write zeros, so the number of the words under consideration is $\binom{n}{k}$. On the other hand, we can first choose $n-k$ places in which to write ones in $\binom{n}{n-k}$ ways, so the number of the words under consideration is $\binom{n}{n-k}$. Therefore, $\binom{n}{k} = \binom{n}{n-k}$.

(b) The algebraic proof of the Pascal's identity is easy:

$$\begin{aligned}\binom{n-1}{k-1} + \binom{n-1}{k} &= \frac{(n-1)!}{(k-1)!(n-k)!} + \frac{(n-1)!}{k!(n-k-1)!} \\ &= \frac{(n-1)!}{(k-1)!(n-k-1)!}\left(\frac{1}{n-k} + \frac{1}{k}\right) \\ &= \frac{(n-1)!}{(k-1)!(n-k-1)!} \cdot \frac{n}{k(n-k)} = \binom{n}{k}\end{aligned}$$

The combinatorial proof uses another important combinatorial principle:

The Sum Principle: *If $A_1, \ldots, A_n$ are mutually disjoint finite sets, then* $|A_1 \cup \ldots \cup A_n| = |A_1| + \ldots + |A_n|$.

Let $S = \{1, 2, \ldots, n\}$ be an n-element set. Clearly, the number of k-element subsets is $\binom{n}{k}$. On the other hand, all k-element subsets of S split into two classes: those that contain 1, and those that do not. The number of k-element subsets of S that contain 1 is $\binom{n-1}{k-1}$ since we have to choose $k-1$ elements from an $(n-1)$-element set $S' = \{2, \ldots, n\}$. The number of k-element subsets of S that do not contain 1 is $\binom{n}{k-1}$ since now we have to choose all k elements from S'. Therefore, by the Sum Principle, $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$. □

Due to the following important result the numbers nk are often referred to as *binomial coefficients*:

Theorem 1.11 (Newton's Binomial Formula) *For all $n \in \mathbb{N}_0$ we have*

$$(a+b)^n = \sum_{k=0}^{n} \binom{n}{k} a^{n-k} b^k.$$

Proof. The proof proceeds by induction on n. The first few cases are trivial:

$$(a+b)^0 = 1 = \binom{0}{0}$$
$$(a+b)^1 = a + b = \binom{1}{0} a + \binom{1}{1} b$$
$$(a+b)^2 = a^2 + 2ab + b^2 = \binom{2}{0} a^2 + \binom{2}{1} ab + \binom{2}{2} b^2$$

Assume that the claim is true for n and let us compute $(a+b)^{n+1}$. By the induction hypothesis:

$$(a+b)^{n+1} = (a+b) \cdot (a+b)^n = (a+b) \cdot \sum_{k=0}^{n} \binom{n}{k} a^{n-k} b^k.$$

After distributing the sum and multiplying we obtain:

$$(a+b)^{n+1} = \sum_{k=0}^{n} \binom{n}{k} a^{n-k+1} b^k + \sum_{k=0}^{n} \binom{n}{k} a^{n-k} b^{k+1}.$$

Next, we take out the first summand in the first sum and the last summand in the second sum to obtain:

$$(a+b)^{n+1} = a^{n+1} + \sum_{k=1}^{n} \binom{n}{k} a^{n-k+1} b^k + \sum_{k=0}^{n-1} \binom{n}{k} a^{n-k} b^{k+1} + b^{n+1}$$

and reindex the second sum, which is a standard trick:

$$(a+b)^{n+1} = a^{n+1} + \sum_{k=1}^{n} \binom{n}{k} a^{n-k+1} b^k + \sum_{m=1}^{n} \binom{n}{m-1} a^{n-m+1} b^m + b^{n+1}.$$

Putting the two sums together we obtain:

$$(a+b)^{n+1} = a^{n+1} + \sum_{k=1}^{n} \left(\binom{n}{k} + \binom{n}{k-1} \right) a^{n-k+1} b^k + b^{n+1}.$$

Finally, we apply the Pascal's identity and wrap it up:

$$(a+b)^{n+1} = a^{n+1} + \sum_{k=1}^{n} \binom{n+1}{k} a^{n-k+1} b^k + b^{n+1} = \sum_{k=0}^{n+1} \binom{n+1}{k} a^{n-k+1} b^k.$$

The combinatorial proof of the Newton's Binomial Formula is based on a simple observation. Clearly,

$$(a+b)^n = \underbrace{(a+b) \cdot (a+b) \cdot \ldots \cdot (a+b)}_{n \text{ times}}$$

so if one multiplies out and writes down the summands as words of length n (that is, without the usuall abbreviations such as $a \cdot a \cdot a = a^3$), one obtains all possible words od length n in letters a and b. For example,

$$\begin{aligned}(a+b)^4 = {} & aaaa + aaab + aaba + aabb + abaa + abab + abba + abbb \\ & + baaa + baab + baba + babb + bbaa + bbab + bbba + bbbb.\end{aligned}$$

There are $\binom{n}{k}$ words that abbreviate to $a^{n-k}b^k$ since this is the number of ways we can choose k places for b (Problem 1.9). Therefore, $a^{n-k}b^k$ appears $\binom{n}{k}$ times in the sum, whence $(a+b)^n = \sum_{k=0}^{n} \binom{n}{k} a^{n-k} b^k$. □

Theorem 1.12 $\binom{n}{0}+\binom{n}{1}+\binom{n}{2}+\ldots+\binom{n}{n}=2^n$.

Proof. For the algebraic proof, just note that

$$2^n=(1+1)^n=\sum_{k=0}^{n}\binom{n}{k}1^{n-k}1^k=\sum_{k=0}^{n}\binom{n}{k}$$

by the Newton's Binomial Formula. The combinatorial proof is also not very complicated. Let A be an arbitraty n-element set and let us count the number of subsets of A. According to Theorem 1.6 this number is 2^n. On the other hand, let us split $\mathscr{P}(A)$ into disjoint collections $\mathscr{S}_0$, $\mathscr{S}_1$, …, $\mathscr{S}_n$ so that $\mathscr{S}_k$ contains all k-element subsets of A. Clearly

$$|\mathscr{P}(A)|=|\mathscr{S}_0|+|\mathscr{S}_1|+\ldots+|\mathscr{S}_n|.$$

But, $|\mathscr{S}_k|=\binom{n}{k}$ according to Theorem 1.8. This concludes the proof. □

Example 1.13 Show that $\dfrac{m^n+(m-2)^n}{2}$ is the number of words of length n over an m-letter alphabet $A=\{a_1,\ldots,a_m\}$ with the additional property that the number of occurences of letter a_1 is even.

Solution. For each even k, $0\leqslant k\leqslant n$, the number of words of length n over A where a_1 occurs k times is $\binom{n}{k}(m-1)^{n-k}$. Therefore, the number of words we are interested in can be expressed as the sum $\sum\limits_{\substack{0\leqslant k\leqslant n\\ k\text{ even}}}\binom{n}{k}(m-1)^{n-k}$.

On the other hand,

$$m^n=((m-1)+1)^n=\sum_{k=0}^{n}\binom{n}{k}(m-1)^{n-k}$$

$$(m-2)^n=((m-1)-1)^n=\sum_{k=0}^{n}(-1)^k\binom{n}{k}(m-1)^{n-k}$$

whence

$$\begin{aligned}m^n+(m-2)^n&=\sum_{k=0}^{n}\left(\binom{n}{k}(m-1)^{n-k}+(-1)^k\binom{n}{k}(m-1)^{n-k}\right)\\&=2\cdot\sum_{\substack{0\leqslant k\leqslant n\\ k\text{ even}}}\binom{n}{k}(m-1)^{n-k}.\end{aligned}$$

This completes the proof.

The Sum Principle states that $|A_1 \cup \ldots \cup A_n| = |A_1| + \ldots + |A_n|$ whenever A_1, $\ldots$, A_n are mutually disjoint finite sets. But, what happens if A_1, $\ldots$, A_n are not mutually disjoint? In case of $n = 2$ we know from the elementary school that

$$|A_1 \cup A_2| = |A_1| + |A_2| - |A_1 \cap A_2|,$$

and it is also easy to see that in case $n = 3$:

$$\begin{aligned} |A_1 \cup A_2 \cup A_3| = {} & |A_1| + |A_2| + |A_3| \\ & - |A_1 \cap A_2| - |A_1 \cap A_3| - |A_2 \cap A_3| \\ & + |A_1 \cap A_2 \cap A_3|. \end{aligned}$$

Theorem 1.14 (The Principle of Inclusion-Exclusion) *Let A_1, $\ldots$, A_n be finite sets. Then*

$$\begin{aligned} |A_1 \cup \ldots \cup A_n| = {} & |A_1| + \ldots + |A_n| \\ & - |A_1 \cap A_2| - |A_1 \cap A_3| - \ldots - |A_{n-1} \cap A_n| \\ & + |A_1 \cap A_2 \cap A_3| + |A_1 \cap A_2 \cap A_4| + \ldots + |A_{n-2} \cap A_{n-1} \cap A_n| \\ & - \ldots \\ & + (-1)^{n-1} |A_1 \cap A_2 \cap \ldots \cap A_n| \end{aligned}$$

Proof. The proof is by induction on n. In case $n = 1$ the formula is trivial and we have already seen that the formula is true in case $n = 2$ or $n = 3$. Therefore, assume that the formula is true in case of n finite sets and let us consider the untion of $n+1$ finite sets. Using the formula for the cardinality of the union of two sets:

$$\begin{aligned} |A_1 \cup \ldots \cup A_n \cup A_{n+1}| &= |(A_1 \cup \ldots \cup A_n) \cup A_{n+1}| \\ &= |A_1 \cup \ldots \cup A_n| + |A_{n+1}| - |(A_1 \cup \ldots \cup A_n) \cap A_{n+1}| \\ &= |A_1 \cup \ldots \cup A_n| + |A_{n+1}| - |(A_1 \cap A_{n+1}) \cup \ldots \cup (A_n \cap A_{n+1})| \end{aligned}$$

the proof follows straightforwardly by applying the induction hypothesis twice. We first apply the induction hypothesis to $|A_1 \cup \ldots \cup A_n|$ and then to $|A'_1 \cup \ldots \cup A'_n|$

where $A_i' = A_i \cap A_{n+1}$:

$$\begin{aligned}
|A_1 \cup \ldots \cup A_n \cup A_{n+1}| = \Big(& |A_1| + \ldots + |A_n| \\
& - |A_1 \cap A_2| - |A_1 \cap A_3| - \ldots - |A_{n-1} \cap A_n| \\
& + |A_1 \cap A_2 \cap A_3| + \ldots + |A_{n-2} \cap A_{n-1} \cap A_n| \\
& - \ldots \\
& + (-1)^{n-1} |A_1 \cap A_2 \cap \ldots \cap A_n| \Big) \\
& + |A_{n+1}| \\
& - \Big(|A_1 \cap A_{n+1}| + \ldots + |A_n \cap A_{n+1}| \\
& \quad - |A_1 \cap A_2 \cap A_{n+1}| - \ldots - |A_{n-1} \cap A_n \cap A_{n+1}| \\
& \quad + \ldots \\
& \quad + (-1)^{n-1} |A_1 \cap A_2 \cap \ldots \cap A_n \cap A_{n+1}| \Big) \\
& = |A_1| + \ldots + |A_n| + |A_{n+1}| \\
& \quad - |A_1 \cap A_2| - |A_1 \cap A_3| - \ldots - |A_n \cap A_{n+1}| \\
& \quad + |A_1 \cap A_2 \cap A_3| + \ldots + |A_{n-1} \cap A_n \cap A_{n+1}| \\
& \quad - \ldots \\
& \quad + (-1)^n |A_1 \cap A_2 \cap \ldots \cap A_n \cap A_{n+1}|,
\end{aligned}$$

which completes the proof. □

Corollary 1.15 *Let $A_1, \ldots, A_n$ be finite sets such that*

$$|A_{i_1} \cap \ldots \cap A_{i_k}| = |A_{j_1} \cap \ldots \cap A_{j_k}|$$

whenever $i_1, \ldots, i_k$ are k distinct indices and $j_1, \ldots, j_k$ are k distinct indices, $k \in \{1, \ldots, n\}$. Then

$$\begin{aligned}
|A_1 \cup \ldots \cup A_n| = & \binom{n}{1} |A_1| - \binom{n}{2} |A_1 \cap A_2| + \binom{n}{3} |A_1 \cap A_2 \cap A_3| - \ldots \\
& + (-1)^{n-1} \binom{n}{n} |A_1 \cap A_2 \cap \ldots \cap A_n|
\end{aligned}$$

Proof. By the Principle of Inclusion-Exclusion:

$$\begin{aligned}|A_1 \cup \ldots \cup A_n| = {} & |A_1| + \ldots + |A_n| \\ & - |A_1 \cap A_2| - |A_1 \cap A_3| - \ldots - |A_{n-1} \cap A_n| \\ & + |A_1 \cap A_2 \cap A_3| + |A_1 \cap A_2 \cap A_4| + \ldots + |A_{n-2} \cap A_{n-1} \cap A_n| \\ & - \ldots \\ & + (-1)^{n-1} |A_1 \cap A_2 \cap \ldots \cap A_n|\end{aligned}$$

The assumption now yields:

$$\begin{aligned}|A_1| + \ldots + |A_n| &= \binom{n}{1} |A_1| \\ |A_1 \cap A_2| + \ldots + |A_{n-1} \cap A_n| &= \binom{n}{2} |A_1 \cap A_2| \\ |A_1 \cap A_2 \cap A_3| + \ldots + |A_{n-2} \cap A_{n-1} \cap A_n| &= \binom{n}{3} |A_1 \cap A_2 \cap A_3| \\ &\ldots \\ (-1)^{n-1} |A_1 \cap A_2 \cap \ldots \cap A_n| &= (-1)^{n-1} \binom{n}{n} |A_1 \cap A_2 \cap \ldots \cap A_n|\end{aligned}$$

which completes the proof. □

A permutation $a_1 \ldots a_n$ of $\{1, \ldots, n\}$ is called a *derangement* if $a_1 \neq 1$, $a_2 \neq 2$, $\ldots$, $a_n \neq n$. For example, 21453 is a derangement of $\{1,2,3,4,5\}$, while 21354 is not. Let D_n denote the number of derangements of $\{1, \ldots, n\}$.

Theorem 1.16 $D_n = n! \cdot \sum_{k=0}^{n} \frac{(-1)^k}{k!}$.

Proof. Let S be the set of all permutations of $\{1, \ldots, n\}$ and let A_j be the set of all permutations $b_1 \ldots b_n \in S$ with $b_j = j$. Then $A_1 \cup \ldots \cup A_n$ is the set of permutations of $\{1, \ldots, n\}$ which are *not* derangements, whence $D_n = |S| - |A_1 \cup \ldots \cup A_n|$. Clearly, $|S| = n!$ while we compute $|A_1 \cup \ldots \cup A_n|$ using the Principle of Inclusion-Exclusion. In order to do so, we have to compute $|A_{i_1} \cap \ldots \cap A_{i_k}|$ for all choices of indices $i_1, \ldots, i_k$ with $i_1 < i_2 < \ldots < i_k$. But this is easy: $A_{i_1} \cap \ldots \cap A_{i_k}$ is the set of all permutations $a_1 \ldots a_n$ from S with the property that $a_{i_1} = i_1$, $a_{i_2} = i_2$, $\ldots$, $a_{i_k} = i_k$, so $|A_{i_1} \cap \ldots \cap A_{i_k}| = (n-k)!$. Using Corollary 1.15 we get

$$|A_1 \cup \ldots \cup A_n| = \sum_{k=1}^{n} (-1)^{k-1} \binom{n}{k} (n-k)!$$

whence

$$D_n = |S| - |A_1 \cup \ldots \cup A_n| = n! - \sum_{k=1}^{n} (-1)^{k-1} \binom{n}{k} (n-k)!$$
$$= \sum_{k=0}^{n} (-1)^k \binom{n}{k} (n-k)!$$

which concludes the proof. □

1.3 Multisets

Two sets are equal if their elements are the same, or more precisely:

$$A = B \quad \text{if and only if} \quad \forall x (x \in A \Leftrightarrow x \in B).$$

As a consequence, $\{b,a,n,a,n,a\} = \{a,b,n\}$. We usually say that "in a set one can omit repeating elements". But what if we *wish* to put several copies of an object into a set? Well, we have to invent a new type of mathematical object.

Definition 1.17 Let $A = \{a_1, a_2, \ldots, a_n\}$ be a finite set. A *multiset over A* is any mapping $\alpha : A \to \mathbb{N}_0^\infty$.

The idea behind this definition is simple: $\alpha(a_k)$ tells us how many copies of a_k we have in the multiset α. This is why α is sometimes called the *multiplicity function*, and $\alpha(a_k)$ is the *multiplicity* of a_k. In particular, $\alpha(a_k) = 0$ means that a_k does not belong to the multiset, while $\alpha(a_k) = \infty$ means that we have an unlimited supply of copies of a_k.

A multiset $\alpha : A \to \mathbb{N}_0^\infty$ can be compactly represented as

$$\alpha = \begin{pmatrix} a_1 & a_2 & \ldots & a_n \\ m_1 & m_2 & \ldots & m_n \end{pmatrix}$$

or, even more conveniently, as

$$\{m_1 \cdot a_1, m_2 \cdot a_2, \ldots, m_n \cdot a_n\},$$

where $m_j = \alpha(a_j)$, $j \in \{1,2,\ldots,n\}$.

Definition 1.18 A multiset $\alpha = \{m_1 \cdot a_1, m_2 \cdot a_2, \ldots, m_n \cdot a_n\}$ is *empty* if $m_1 = \ldots = m_n = 0$. The multiset α is *finite* if $m_1, \ldots, m_n < \infty$. The number of elements of α is denoted by $|\alpha|$ and we define it by $|\alpha| = \sum_{a \in A} \alpha(a)$.

A multiset $\alpha = \{m_1 \cdot a_1, m_2 \cdot a_2, \ldots, m_n \cdot a_n\}$ is a *submultiset* of a multiset $\beta = \{k_1 \cdot a_1, k_2 \cdot a_2, \ldots, k_n \cdot a_n\}$ if $m_j \leqslant k_j$ for all j.

Example 1.19 Let $A = \{a, b, c\}$. Then $\alpha = \{3 \cdot a, 2 \cdot b, 1 \cdot c\}$ and $\beta = \{0 \cdot a, 5 \cdot b, \infty \cdot c\}$ are two multisets over A. Clearly α is a finite multiset with 6 elements, while β is infinite and $|\beta| = \infty$. Both α and β are submultisets of $\gamma = \{\infty \cdot a, 5 \cdot b, \infty \cdot c\}$. Also, β is a submultiset of $\delta = \{1 \cdot a, \infty \cdot b, \infty \cdot c\}$, while α is not.

A *word over a multiset* $\alpha = \{m_1 \cdot a_1, m_2 \cdot a_2, \ldots, m_n \cdot a_n\}$ is any word w over $A = \{a_1, \ldots, a_n\}$ such that $|w|_{a_j} \leqslant m_j$ for all j.

Example 1.20 Let $\alpha = \{3 \cdot a, 2 \cdot b, 2 \cdot n\}$. The following are some words over α: *banana, abba, aa,* but *abbba* is not. As another example, take $\beta = \{1 \cdot a, \infty \cdot b\}$. Then all these are words over β: *a, ab, abb, abbb,* and so on.

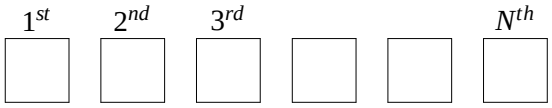
Problem 1.21 *Let $\alpha = \{m_1 \cdot a_1, m_2 \cdot a_2, \ldots, m_n \cdot a_n\}$ be a multiset and let $k \in \mathbb{N}_0$ be arbitrary.*

(a) Suppose $m_1 = m_2 = \ldots = m_n = \infty$. How many words with k letters over α are there?

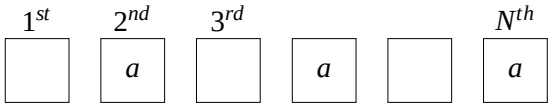
(b) Suppose α is finite. How many words w over α have the property that $|w|_{a_j} = m_j$ for all j?

Solution. (a) Since each letter comes in more than sufficiently many copies, it turns out that the number of such words is n^k. Compare with Problem 1.3 (a).

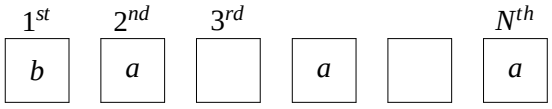
(b) Let $N = |\alpha| = m_1 + \ldots + m_n$. Then the words we are interested are of length N:



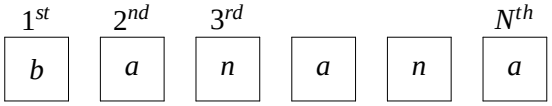
and each letter a_j occurs precisely m_j times. Let us now distribute the letters from α. Out of N free places we can choose m_1 places to put the copies of a_1 in $\binom{N}{m_1}$ ways:



Out of $N - m_1$ remaining free places we can choose m_2 places to put the copies of a_2 in $\binom{N-m_1}{m_2}$ ways:



Out of $N-m_1-m_2$ remaining free places we can choose m_3 places to put the copies of a_3 in $\binom{N-m_1-m_2}{m_3}$ ways, and so on. At the end, out of $N-m_1-m_2-\ldots-m_{n-1}$ remaining free places we can choose m_n places to put the copies of a_n in $\binom{N-m_1-m_2-\ldots-m_{n-1}}{m_n}$ ways:



Therefore, the number of words we are interested in is given by

$$\binom{N}{m_1}\cdot\binom{N-m_1}{m_2}\cdot\binom{N-m_1-m_2}{m_3}\cdot\ldots\cdot\binom{N-m_1-m_2-\ldots-m_{n-1}}{m_n}=$$

$$=\frac{N!}{m_1!(N-m_1)!}\cdot\frac{(N-m_1)!}{m_2!(N-m_1-m_2)!}\cdot\ldots\cdot\frac{(N-m_1-m_2-\ldots-m_{n-1})!}{m_n!(N-m_1-m_2-\ldots-m_n)!}=$$

$$=\frac{N!}{m_1!\cdot m_2!\cdot\ldots\cdot m_n!},$$

where at the end we use the fact that $N=m_1+m_2+\ldots+m_n$. □

A *permutation of a finite multiset* $\alpha=\{m_1\cdot a_1, m_2\cdot a_2,\ldots,m_n\cdot a_n\}$ is any word w over α such that $|w|_{a_j}=m_j$ for all j. As we have just seen, the number of permutations over a finite multiset α is

$$\binom{N}{m_1,m_2,\ldots,m_n}=\frac{N!}{m_1!\cdot m_2!\cdot\ldots\cdot m_n!}$$

where $N=m_1+m_2+\ldots+m_n$. Finding the number of k-letter words for arbitrary k and over an arbitrary multiset is a *terribly* complicated problem and shall not be discussed here.

We shall now prove an analogon on the Newton's Binomial Formula in case a sum of more than two expressions is raised to a certain power.

Theorem 1.22 (Multinomial Formula) *For all $n\geqslant 0$ we have*

$$(a_1+a_2+\ldots+a_k)^n \quad=\sum_{\substack{l_1,l_2,\ldots,l_k\in\mathbb{N}_0\\ l_1+l_2+\ldots+l_k=n}}\binom{n}{l_1,l_2,\ldots,l_k}a_1^{l_1}a_2^{l_2}\ldots a_k^{l_k}$$

Proof. The proof proceeds by induction on k. In case $k = 2$ this is just the Newton's Binomial Formula given in Theorem 1.11, see Homework 1.12. Suppose the theorem holds whenever there are less than k summands whose sum we wish to raise to the n-th power and consider the case with k summands. Then by the Newton's Binomial Formula

$$(a_1 + a_2 + \ldots + a_k)^n = (a_1 + (a_2 + \ldots + a_k))^n = \sum_{l_1=0}^{n} \binom{n}{l_1} a_1^{l_1} (a_2 + \ldots + a_k)^{n-l_1}.$$

The induction hypothesis now yields

$$\begin{aligned}
(a_1 + a_2 + \ldots + a_k)^n &= \sum_{l_1=0}^{n} \binom{n}{l_1} a_1^{l_1} \sum_{\substack{l_2,\ldots,l_k \in \mathbb{N}_0 \\ l_2+\ldots+l_k = n-l_1}} \binom{n-l_1}{l_2,\ldots,l_k} a_2^{l_2} \ldots a_k^{l_k} \\
&= \sum_{l_1=0}^{n} \sum_{\substack{l_2,\ldots,l_k \in \mathbb{N}_0 \\ l_2+\ldots+l_k = n-l_1}} \binom{n}{l_1} \binom{n-l_1}{l_2,\ldots,l_k} a_1^{l_1} a_2^{l_2} \ldots a_k^{l_k} \\
&= \sum_{l_1=0}^{n} \sum_{\substack{l_2,\ldots,l_k \in \mathbb{N}_0 \\ l_2+\ldots+l_k = n-l_1}} \binom{n}{l_1,l_2,\ldots,l_k} a_1^{l_1} a_2^{l_2} \ldots a_k^{l_k} \\
&= \sum_{\substack{l_1,l_2,\ldots,l_k \in \mathbb{N}_0 \\ l_1+l_2+\ldots+l_k = n}} \binom{n}{l_1,l_2,\ldots,l_k} a_1^{l_1} a_2^{l_2} \ldots a_k^{l_k}.
\end{aligned}$$

The combinatorial proof is analogous to the combinatorial proof of Theorem 1.11. □

Problem 1.23 *Let $\alpha = \{\infty \cdot a_1, \infty \cdot a_2, \ldots, \infty \cdot a_n\}$ be a multiset and let $k \in \mathbb{N}_0$ be arbitrary. How many k-element submultisets does α have?*

Solution. If $\beta = \{x_1 \cdot a_1, x_2 \cdot a_2, \ldots, x_n \cdot a_n\}$ is a k-element submultiset of α, then $x_1 + x_2 + \ldots + x_n = k$. Because α has an infinite supply of each of its letters, one easily comes to the following conclusion:

Number of k-element submultisets of α	=	Number of solutions of $x_1 + x_2 + \ldots + x_n = k$ in $\mathbb{N}_0$

So, we have reduced the problem to counting nonnegative integer solutions of an equation in n unknowns. Although not at all straightforward, this problem is rather easy to solve. Let

$$\mathscr{S} = \{(x_1, x_2, \ldots, x_n) \in (\mathbb{N}_0)^n : x_1 + x_2 + \ldots + x_n = k\}$$

be the set of all the solutions of the above equation in n unknowns and let

$$\mathscr{W} = \{w \in \{0,1\}^{k+n-1} : |w|_0 = k \text{ and } |w|_1 = n-1\}$$

be the set of all 01-words of length $k+n-1$ with precisely k zeros and $n-1$ ones. Now define $\varphi : \mathscr{S} \to \mathscr{W}$ as follows:

$$\varphi(x_1, x_2, \ldots, x_n) = \underbrace{00\ldots0}_{x_1}\ 1\ \underbrace{00\ldots0}_{x_2}\ 1\ \ldots\ 1\ \underbrace{00\ldots0}_{x_n}.$$

It is easy to see that φ is well defined and bijective. Therefore, $|\mathscr{S}| = |\mathscr{W}|$, and we know from Problem 1.9 that $|\mathscr{W}| = \binom{k+n-1}{k}$. This is at the same time the number of k-element submultisets of α. □

A *k-combination of a finite multiset α* is any k-element subset of α. It is again *terribly* complicated to find a number of k-combinations of an arbitrary multiset, but as we have just seen, if $\alpha = \{\infty \cdot a_1, \infty \cdot a_2, \ldots, \infty \cdot a_n\}$, the number of k-combinations is given by $\binom{k+n-1}{k}$.

Homework

1.1. For a real number x, by $\lfloor x \rfloor$ we denote the greatest integer $\leqslant x$. E.g, $\lfloor 1.99 \rfloor = 1$, $\lfloor 4 \rfloor = 4$, $\lfloor 0.65 \rfloor = 0$, while $\lfloor -1.02 \rfloor = -2$.

Let n be an integer and p a prime. Show that the greatest k such that $p^k \mid n!$ is given by

$$k = \left\lfloor \frac{n}{p} \right\rfloor + \left\lfloor \frac{n}{p^2} \right\rfloor + \left\lfloor \frac{n}{p^3} \right\rfloor + \ldots$$

The number 1000! ends with a lot of zeros. How many?

1.2. Show that χ in proof of Theorem 1.6 is a bijection.

1.3. Let A be a set of all 01-words w of length 2005 with the property that $|w|_0 = |w|_1 + 1$, and let B be a set of all 01-words w of length 2005 with the property that $|w|_1 = |w|_0 + 1$. Show that $|A| = |B|$. (Hint: use the Bijection Principle.)

1.4. For $n \in \mathbb{N}$, let $\tau(n)$ denote the number of positive divisors of n. E.g, $\tau(12) = 6$ since 1, 2, 3, 4, 6 and 12 are all positive divisors of 12. Let $n = p_1^{k_1} \cdot p_2^{k_2} \cdot \ldots \cdot p_s^{k_s}$ be the factorisation of n, where $1 < p_1 < p_2 < \ldots < p_s$ are primes. Prove that

$$\tau(n) = (1+k_1)(1+k_2)\ldots(1+k_s).$$

(Hint: note that if $m \mid n$ then $m = p_1^{l_1} \cdot p_2^{l_2} \cdot \ldots \cdot p_s^{l_s}$ where $0 \leqslant l_i \leqslant k_i$ for all i.)

1.5. Show that

(a) $\sum_{k=0}^{n} 2^k \binom{n}{k} = 3^n$;

(b) $\binom{n}{0} + \binom{n}{2} + \binom{n}{4} + \ldots = \binom{n}{1} + \binom{n}{3} + \binom{n}{5} + \ldots$;

(c) $\sum_{k=0}^{n} k \binom{n}{k} = n \cdot 2^{n-1}$.

1.6. (a) Find $\lim_{n\to\infty} \frac{D_n}{n!}$.

(b) Show that $D_n = n \cdot D_{n-1} + (-1)^n$, $n \geqslant 2$.

(c) Show that $D_n = (n-1) \cdot (D_{n-1} + D_{n-2})$, $n \geqslant 3$. (Hint: Use the fact that $n! = (n-1) \cdot ((n-1)! + (n-2)!)$; this is why the numbers D_n are sometimes referred to as *subfactorials*.)

1.7. Let $b_1 \ldots b_n$ be a permutation of an n-element set A. Find the number of permutations $a_1 \ldots a_n$ of A having the property that $a_1 \neq b_1$, $a_2 \neq b_2$, …, $a_n \neq b_n$.

1.8. Show that φ defined in the solution to Problem 1.23 is a bijection.

†**1.9.** What do you think, how do "usual" sets fit into the theory of multisets?

†**1.10.** Define the notion of union and intersection for multisets. (Note that there are several possibilities; choose any one you like). Pick a few of your favourite set-theory identities such as

$$\begin{array}{ll} \alpha \cap \alpha = \alpha & \alpha \cup \alpha = \alpha \\ \alpha \cap \varnothing = \varnothing & \alpha \cup \varnothing = \alpha \\ \alpha \cap \beta = \beta \cap \alpha & \alpha \cup \beta = \beta \cup \alpha \\ (\alpha \cap \beta) \cap \gamma = \alpha \cap (\beta \cap \gamma) & (\alpha \cup \beta) \cup \gamma = \alpha \cup (\beta \cup \gamma) \\ (\alpha \cap \beta) \cup \gamma = (\alpha \cup \gamma) \cap (\beta \cup \gamma) & (\alpha \cup \beta) \cap \gamma = (\alpha \cap \gamma) \cup (\beta \cap \gamma) \end{array}$$

and show that they hold for operations you have defined.

1.11. Find the number of k-element subsets of $\{1, \ldots, n\}$ which do not contain adjacent numbers. For example, $\{1, 5, 7, 13\}$ is a good subset of $\{1, \ldots, 15\}$, while $\{2, 4, 6, 7\}$ is not.

1.12. (a) Explain the relationship between $\binom{n}{k}$ and $\binom{n}{k, n-k}$.

(*b*) Show that $\binom{n}{k,n-k} = \binom{n-1}{k-1,n-k} + \binom{n-1}{k,n-k-1}$ (Hint: this is the Pascal's identity in disguise.)

1.13. Let $m_1,\ldots,m_n \in \mathbb{N}$ be positive integers and let $N = m_1 + \ldots + m_n$. Show that

$$\binom{N}{m_1,m_2,\ldots,m_n} = \binom{N-1}{m_1-1,m_2,\ldots,m_n} + \binom{N-1}{m_1,m_2-1,\ldots,m_n} + \ldots$$
$$\ldots + \binom{N-1}{m_1,m_2,\ldots,m_n-1}.$$

1.14. Provide a combinatorial proof of Theorem 1.22.

Exercises

1.15. How much memory can address a processor whose address bus is 32 bits wide?

1.16. FORTRAN IV, being one of the oldest programming languages, had many limitations. One of them concerned identifiers (words used to name variables and procedures). An identifier in FORTRAN IV consists of at most 6 symbols, where each symbol is a figure (0, 1, ..., 9) or an uppercase letter of the English alphabet (A, B, ..., Z), with the exception that the first symbol is obliged to be a letter. How many different identifiers can one declare in FORTRAN IV?

†**1.17.** Show that there are infinitely many triples of positive integers (m,n,k) with the property that $m! \cdot n! = k!$ and $m,n,k \geqslant 2$.

1.18. Let $A = \{n \in \mathbb{N} : 1 \leqslant n \leqslant 999\,999 \text{ and the sum of digits of } n \text{ is } 20\}$, and $B = \{n \in \mathbb{N} : 1 \leqslant n \leqslant 999\,999 \text{ and the sum of digits of } n \text{ is } 34\}$. Show that $|A| = |B|$. (Hint: use the Bijection Principle.)

1.19. Two rooks on a chess board are said to be independent if they do not attack each other. In how many different ways can one arrange $n \geqslant 1$ independent identical rooks onto an $n \times n$ chess board?

1.20. In how many different ways can one arrange $k \geqslant 1$ independent identical rooks onto an $n \times m$ chess board, where $n,m \geqslant k$?

1.21. In how many ways can n students form a queue in front of a mensa so that students A and B

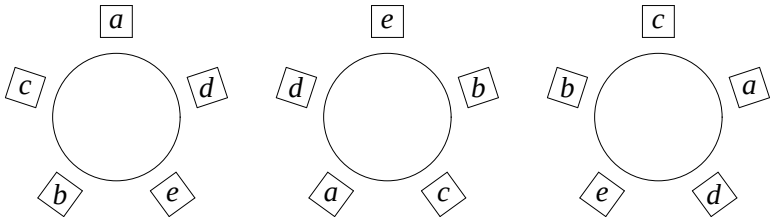
(*a*) are next to each other in the queue?

(b) are *not* next to each other in the queue?

†**1.22.** In how many ways can n boys $B_1, \ldots, B_n$ and n girls $G_1, \ldots, G_n$ form a queue in front of a mensa so that B_1 is next to G_1 in the queue, B_2 is next to G_2 in the queue, $\ldots$, B_n is next to G_n in the queue?

1.23. Find the numbers of pairs (A,B) of subsets of $\{1,\ldots,n\}$ satisfying $A \cap B = \varnothing$.

1.24. The round table has entered combinatorial practice at the time of King Arthur and his Knights of the Round Table and has remained an important combinatorial object ever since. Since there is no throne, the trick with the round table is that two arrangements are indistinguishable if it is possible to get one of them by rotating the other. For example, the following three arrangements are indistinguishable:

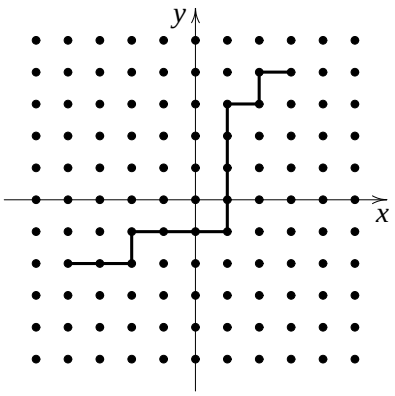


In how many ways can n people be seated around a round table with n seats?

1.25. The *integer grid* consists of all points in the plane with integer coordinates, which we refer to as *integer points*.

An *increasing path* in the integer grid is a sequence of integer points (x_1,y_1), (x_2,y_2), $\ldots$, (x_k,y_k) such that for each $i \in \{1,\ldots,k-1\}$ we have:

- either $x_{i+1} = x_i + 1$ and $y_{i+1} = y_i$,
- or $x_{i+1} = x_i$ and $y_{i+1} = y_i + 1$.



Find the number of increasing paths in the integer grid that start at $(0,0)$ and end at (p,q), where $p,q \in \mathbb{N}$.

1.26. Show that among any 10 distinct points chosen within a square of side 3 one can find two whose distance is $\leqslant \sqrt{2}$.

1.27. Is it possible to fill the entries of an $n \times n$ table with integers -1, 0 and 1 so that the sums of each row, each column and both diagonals are all distinct?

1.28. Show that

(a) $\binom{n}{k} = \frac{n}{k}\binom{n-1}{k-1}$ for all $n \geqslant k \geqslant 1$;

(b) $\binom{n}{m}\binom{m}{k} = \binom{n}{k}\binom{n-k}{m-k}$ for all $n \geqslant m \geqslant k \geqslant 0$;

(c) $\binom{n}{0}\binom{m}{k} + \binom{n}{1}\binom{m}{k-1} + \binom{n}{2}\binom{m}{k-2} + \ldots + \binom{n}{k}\binom{m}{0} = \binom{n+m}{k}$ for all $n, m \geqslant k \geqslant 0$.

(d) $\binom{n}{0}^2 + \binom{n}{1}^2 + \binom{n}{2}^2 + \ldots + \binom{n}{n}^2 = \binom{2n}{n}$ for all $n \geqslant 0$.

(e) $\binom{k}{0} + \binom{k+1}{1} + \binom{k+2}{2} + \ldots + \binom{k+j}{j} = \binom{k+j+1}{j}$ for all $k, j \geqslant 0$. (Hint: use mathematical induction on j.)

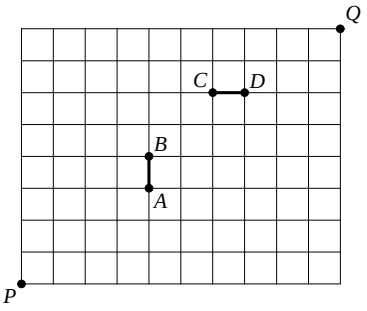
1.29. Find the number of 01-words of length $2n$ which have the following property: the number of zeros on the first n places equals the number of zeros on the last n places.

1.30. (a) Using the fact that two points determine precisely one straingt line, find the greatest number of straignt lines that can be drawn through n points in a plane.

(b) Find the greatest number of diagonals a convex polygon with n vertices can have.

(c) Let $A_1, \ldots, A_n$ be n points on a circle, $n \geqslant 4$, and draw all the line segments A_iA_j, $i \neq j$. Find the greatest possible number of intersection points of these line segments.

1.31. Find the number of increasing paths (in the integer grid) which go from P to Q and avoid line segments AB and CD.



1.32. Find the number of permutations $a_1a_2a_3a_4a_5$ of $\{1,2,3,4,5\}$ such that $|a_1-a_2|\neq 1$, $|a_2-a_3|\neq 1$, $|a_3-a_4|\neq 1$ and $|a_4-a_5|\neq 1$.

1.33. Show that $\sum_{k=0}^{n}(-1)^k\binom{n}{k}(n-k)^n = n!$. (Hint: Using Double Counting and the Principle of Inclusion-Exclusion find the number of words of length n over an n-letter alphabet where each letter from the alphabet appears in the word.)

1.34. (a) Find the number of integer solutions of the equation

$$x_1+x_2+\ldots+x_n = k$$

in n unknowns $x_1, x_2, \ldots, x_n$ where $x_i \geqslant 1$ for all i.

(b) Find the number of integer solutions of the equation

$$x_1+x_2+\ldots+x_{k+1} = n-k$$

in $k+1$ unknowns $x_1, x_2, \ldots, x_{k+1}$ where $x_1 \geqslant 0$, $x_{k+1}\geqslant 0$ and $x_i \geqslant 1$ for all $i \in \{2,\ldots,k\}$.

1.35. Find the number of integer solutions of the inequality $x_1+x_2+\ldots+x_n \leqslant k$ in n unknowns $x_1, x_2, \ldots, x_n$ where $x_i \geqslant 0$ for all i. (Hint: Since $k\in\mathbb{N}_0$, this inequality is equivalent to

$$x_1+x_2+\ldots+x_n = 0 \quad \text{or} \quad x_1+x_2+\ldots+x_n = 1 \quad \text{or} \ \ldots$$
$$\ldots \ \text{or} \quad x_1+x_2+\ldots+x_n = k.$$

Find the number of solutions of each of these $k+1$ equations and then sum up using 1.28 (e).)

1.36. An integer solution $(x_1,\ldots,x_n)$ of the equation $x_1+x_2+\ldots+x_n = k$, $k\geqslant 1$, in n unknowns $x_1, x_2, \ldots, x_n$ is called *even* if x_1 is even. Otherwise it is called *odd*. Show that the number of even solutions is greater than the number of odd solutions, provided that $n\geqslant 3$. (Hint: Show that $\varphi : (x_1,x_2,x_3,\ldots,x_n)\mapsto(x_1-1,x_2+1,x_3,\ldots,x_n)$ is an injective mapping from the set of odd solutions into the set of even solutions and note that there exists an even solution which is not in the image of φ.)

1.37. Find the number of integer solutions of the equation

$$x+y+z \equiv 0 \pmod 3$$

where $x,y,z\in\{1,2,3,\ldots,3n\}$.

1.38. A sequence of numbers $x_1, x_2, \ldots, x_n$ is nondecreasing if $x_1 \leqslant x_2 \leqslant \ldots \leqslant x_n$. Find the number of nondecreasing sequences $x_1, x_2, \ldots, x_n$ where $x_i \in \{1, \ldots, k\}$ for all i.

1.39. There are n knights sitting around a round table. Find the number of ways to choose k of those n knights in such a way that no two of the chosen k knights are sitting next to one another. (Hint: pick a knight A and then split all the choices into two disjoint classes – those where A takes part, and those where A does not.)

1.40. Find the number of n-digit positive integers ($n \geqslant 2$) whose sum of digits is 11.

1.41. Show that $\displaystyle\sum_{\substack{l_1, l_2, \ldots, l_k \in \mathbb{N}_0 \\ l_1 + l_2 + \ldots + l_k = n}} \binom{n}{l_1, l_2, \ldots, l_k} = k^n$.

Chapter 2

Graphs and Digraphs

Graphs represent one of the most popular tools for modeling discrete phenomena where the abstraction of the problem involves information about certain objects being connected or not. For example, crossings in a city transportation model are joined by streets, or cities in a country are joined by roads. We will examine two types of such models: graphs which correspond to situations where all the "roads" are bidirectional, and digraphs (*di*rected *graphs*) where one-way "roads" are allowed.

2.1 Graphs

A *graph* is an ordered pair $G = (V,E)$ where V is a nonempty finite set and E is an arbitrary subset of $V^{(2)} = \big\{\{u,v\} \subseteq V : u \neq v\big\}$. Elements of V are called *vertices* of G, while elements of E are called *edges* of G. We shall often write $V(G)$ and $E(G)$ to denote the set of vertices and the set of edges of G, and $n(G)$ and $m(G)$ to denote the number of vertices and the number of edges of G. If $e = \{u,v\}$ is an edge of a graph, we say that u and v are *adjacent*, and that e is *incident* with u and v. We also say that u is a *neighbour* of v. The *neighbour-set of v* is the set $N_G(v) = \{x \in V(G) : x \text{ is a neighbour of } v\}$. The *degree of a vertex v*, denoted by $\delta_G(v)$, is the number of edges incident to v: $\delta_G(v) = |N_G(v)|$. If G is clear from the context, we simply write $N(v)$ and $\delta(v)$. By $\delta(G)$ we denote the least, and by $\Delta(G)$ the greatest degree of a vertex in G. A vertex with degree 0 is said to be an *isolated vertex*. A vertex of degree 1 is called a *leaf of G*. A vertex is said to be *even*, resp. *odd* according as $\delta(v)$ is an even or an odd integer. A graph is *regular* if $\delta(G) = \Delta(G)$. In other words, in a regular graph all vertices have the same degree.

The graphs are called graphs because of a very natural graphical representation they have. Vertices are usually represented as (somewhat larger) points in a plane,

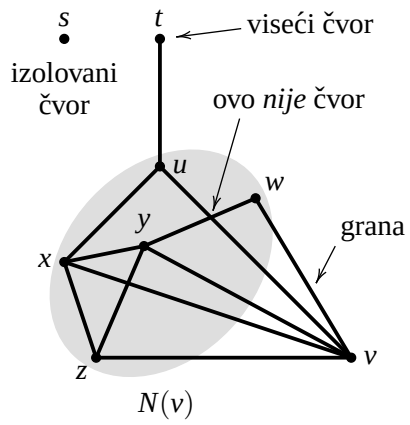


Figure 2.1: An example of a graph

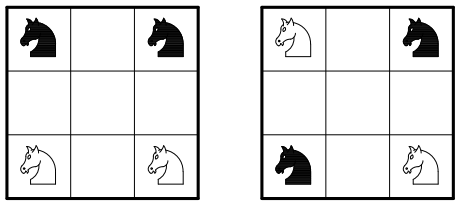
while edges are represented as (smooth non-selfintersecting) curves joining the respective vertices, so that adjacent vertices are joined by a curve.

Example 2.1 Fig. 2.1 depicts a graph G with $V = \{s,t,u,v,w,x,y,z\}$ and $E = \big\{\{t,u\}, \{u,x\}, \{u,v\}, \{w,y\}, \{w,v\}, \{v,x\}, \{v,y\}, \{v,z\}, \{x,y\}, \{x,z\}, \{y,z\}\big\}$. We see that

vertex	s	t	u	v	w	x	y	z
δ	0	1	3	5	2	4	4	3

so $\delta(G) = 0$ and $\Delta(G) = 5$. Also, $N(v) = \{u,w,x,y,z\}$.

Example 2.2 Two black and two white knights are placed on a 3×3 chessboard as in Fig. 2.2 (a). Is it possible to reach the configuration in Fig. 2.2 (b) following the rules of chess?



(a) (b)

Figure 2.2: Example 2.2

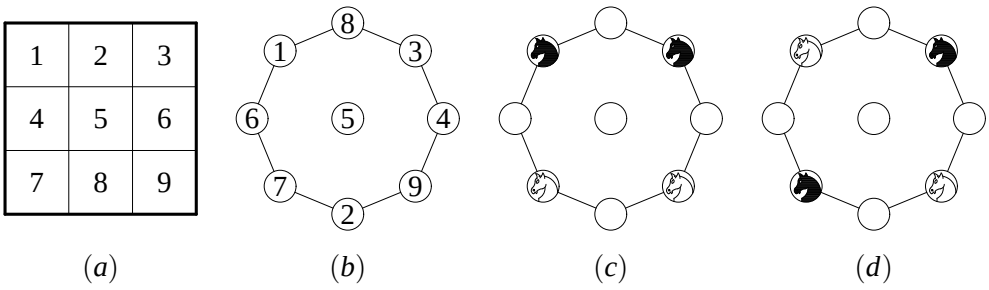


Figure 2.3: Solution to the problem in Example 2.2

Answer: No. Let us enumerate the fields of the chess board by 1, ..., 9 as in Fig. 2.3 (a). To this chess board we can now assign a graph with $\{1,\ldots,9\}$ as the set of vertices by joining i and j if an only if it is possible for a knight to jump from i to j following the general rules of chess. The graph is given in Fig. 2.3 (b). Clearly, regular movements of a knight on the 3×3 chess board correspond to movements of the knight along the edges of the graph in Fig. 2.3 (b). We see now that it is not possible to start from the initial position of the knights given in Fig. 2.3 (c) and reach the final position in Fig. 2.3 (d) by moving one knight at a time along the edges of the graph simply because the white knights separate the black knights in Fig. 2.3 (d), which is not the case in the initial position.

Theorem 2.3 (The First Theorem of Graph Theory) *If $G=(V,E)$ is a graph with m edges, then $\sum_{v\in V}\delta(v)=2m$.*

Proof. Since every edge is incident to two vertices, every edge is counted twice in the sum on the left. □

Corollary 2.4 *In any graph the number of odd vertices is even.*

Theorem 2.5 *If $n(G)\geqslant 2$, there exist vertices $v,w\in V(G)$ such that $v\neq w$ and $\delta(v)=\delta(w)$.*

Proof. Let $V(G)=\{v_1,\ldots,v_n\}$ and suppose that $\delta(v_i)\neq\delta(v_j)$ whenever $i\neq j$. Without loss of generality we may assume that $\delta(v_1)<\delta(v_2)<\ldots<\delta(v_n)$. Since there are only n possibilities for the degree of a vertex (0, 1, ..., $n-1$) it follows that $\delta(v_1)=0$, $\delta(v_2)=1$, ..., $\delta(v_n)=n-1$. But then v_n is adjacent to every other vertex of a graph, including the isolated vertex v_1. Contradiction. □

A graph $H=(W,E')$ is a *subgraph* of a graph $G=(V,E)$, in symbols $H\leqslant G$, if $W\subseteq V$ and $E'\subseteq E$. A subgraph H of G is a *spanning subgraph* if $W=V(G)$.

A subgraph H is an *induced subgraph* of G if $E' = E \cap W^{(2)}$. Induced subgraphs are usually denoted by $G[W]$. The edges of an induced subgraph of G are all the edges of G whose both ends are in W. A set of vertices $W \subseteq V(G)$ is *independent* if $E(G[W]) = \varnothing$, i.e. no two vertices in W are adjacent in G. If $A, B \subseteq E(G)$ are disjoint, by $E(A,B)$ we denote the set of all edges in G whose one end is in A and the other in B.

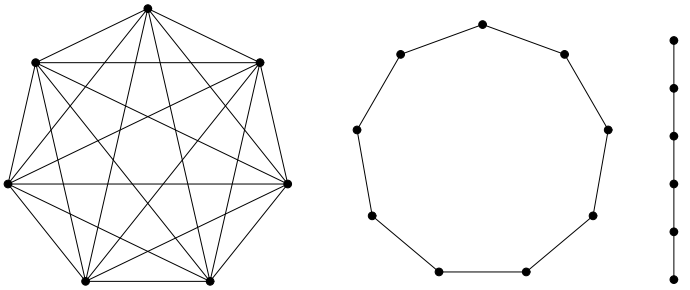


Figure 2.4: K_7, C_9 and P_6

A *complete graph on n vertices* (or an *n-clique*) is a graph with n vertices where each two distinct vertices are adjacent. A complete graph on n vertices is denoted by K_n. A *cycle* of length n, denoted by C_n, is the graph with n vertices where the first vertex is adjacent to the second one, and the second vertex to the third one, and so on, the last vertex is adjacent to the first. A *path with n vertices*, denoted by P_n, is a graph where the first vertex is adjacent to the second one, and the second vertex to the third one, and so on, and the penultimate vertex is adjacent to the last one, but the last vertex is *not* adjacent to the first. We say that the path with n vertices has length $n-1$. Fig. 2.4 depicts K_7, C_9 and P_6.

Theorem 2.6 *If $\delta(G) \geqslant 2$ then G contains a cycle.*

Proof. Let $x_1 \ldots x_{k-1}\, x_k$ be the longest path in G. Since $\delta(x_k) \geqslant \delta(G) \geqslant 2$, x_k has a neighbour v distinct from x_{k-1}. If $v \notin \{x_1, \ldots, x_{k-2}\}$ then $x_1 \ldots x_{k-1}\, x_k\, v$ is a path with more vertices than the longest path, which is impossible. Therefore, $v = x_j$ for some $j \in \{1, \ldots, k-2\}$ so $x_j \ldots x_k$ are vertices of a cycle in G. □

Graphs G_1 and G_2 are *isomorhic*, and we write $G_1 \cong G_2$, if there is a bijection $\varphi : V(G_1) \to V(G_2)$ such that $\{x,y\} \in E(G_1) \Leftrightarrow \{\varphi(x), \varphi(y)\} \in E(G_2)$. For example graphs G and G_2 in Fig. 2.5 are isomorphic, while G and G_1 are not.

Theorem 2.7 *Let $G_1 \cong G_2$ and let φ be an isomorphism between G_1 and G_2. Then $n(G_1) = n(G_2)$, $m(G_1) = m(G_2)$ and $\delta_{G_1}(x) = \delta_{G_2}(\varphi(x))$ for every $x \in V(G_1)$.*

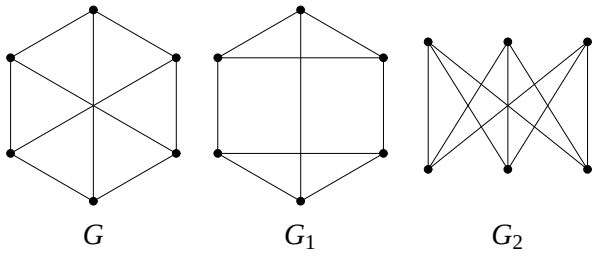


Figure 2.5: $G \cong G_2$, but $G \ncong G_1$

The *complement* of a graph $G = (V,E)$ is the graph $\overline{G} = (V,\overline{E})$ where $\overline{E} = V^{(2)} \setminus E$. A graph G is *selfcomplementary* if $G \cong \overline{G}$. Clearly, $m(G) + m(\overline{G}) = \binom{n}{2}$.

Lemma 2.8 *Let G and H be graphs.*
(a) $G \cong H$ if and only if $\overline{G} \cong \overline{H}$.
(b) $\delta_{\overline{G}}(x) = (n(G)-1) - \delta_G(x)$ for all $x \in V(G)$.

Theorem 2.9 *If G is a selfcomplementary graph with n vertices then $n \geqslant 4$ and $n \equiv 0,1 \pmod 4$. Conversely, for every integer $n \geqslant 4$ such that $n \equiv 0,1 \pmod 4$ there exists a selfcomplementary graph with n vertices.*

Proof. Let G be a selfcomplementary graph with $n \geqslant 4$ vertices and m edges and let $\overline{m} = m(\overline{G})$. Then $m + \overline{m} = \binom{n}{2}$ and $m = \overline{m}$ since $G \cong \overline{G}$. Therefore $2m = \dfrac{n(n-1)}{2}$ i.e. $m = \dfrac{n(n-1)}{4}$. But m is an integer and n and $n-1$ are not of the same parity, so $4 \mid n$ or $4 \mid n-1$.

For the other part of the statement, for every integer $n \geqslant 4$ such that $n \equiv 0,1 \pmod 4$ we shall construct a selfcomplementary graph $G_n = (V_n, E_n)$ with n vertices. It is obvious that we can take $G_4 = P_4$ and $G_5 = C_5$. Now let G_n be a selfcomplementary graph with n vertices and construct G_{n+4} as follows. Take four new vertices t, u, v, w and put

$$V_{n+4} = V_n \cup \{t,u,v,w\}$$
$$E_{n+4} = E_n \cup \{\{t,u\},\{u,v\},\{v,w\}\} \cup \{\{t,x\} : x \in V_n\} \cup \{\{w,x\} : x \in V_n\},$$

see Fig. 2.6 (a). Then $\overline{G}_{n+4}$ is given in Fig. 2.6 (b) and it is easy to establish that $G_{n+4} \cong \overline{G}_{n+4}$. □

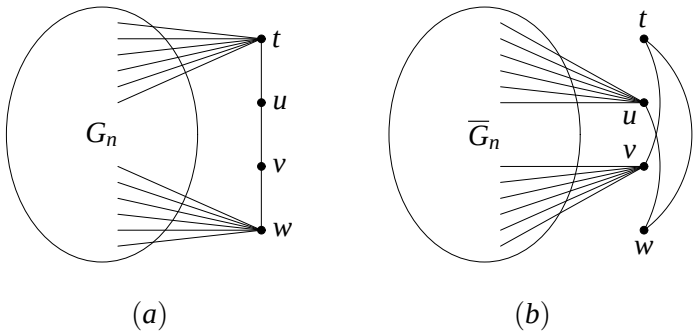


Figure 2.6: The proof of Theorem 2.9

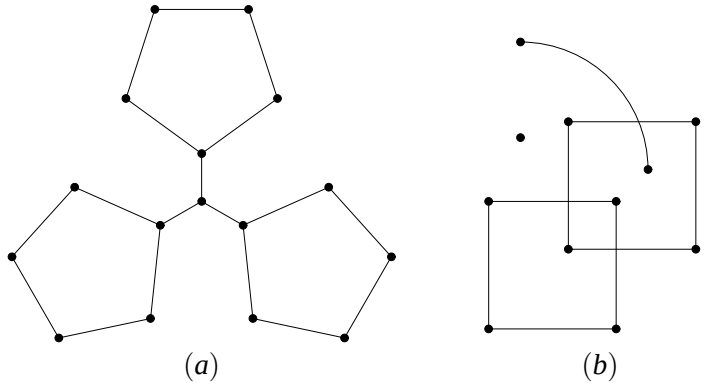


Figure 2.7: (a) A connected graph; (b) A graph with $\omega = 4$

2.2 Connectedness and distance

A *walk* in a graph G is any sequence of vertices and edges $v_0\, e_1\, v_1\, e_2\, v_2 \ldots v_{k-1}\, e_k\, v_k$ such that $e_i = \{v_{i-1}, v_i\}$ for all $i \in \{1, \ldots, k\}$. Note that an edge or a vertex may appear more than once in a walk. We say that k is the *length* of the walk. If $v_0 \neq v_k$ we say that the *walk connects* v_0 *and* v_k. A *closed walk* is a walk $v_0\, e_1\, v_1\, \ldots\, v_{k-1}\, e_k\, v_k$ where $v_0 = v_k$. Clearly, a path is a walk where neither vertices nor edges are allowed to repeat, and a cycle is a closed walk where neither edges nor vertices are allowed to repeat, except for the first and the last vertex.

Lemma 2.10 *If there is a walk in G that connects two vertices then there is a path that connects them. Every closed walk of odd length contains an odd cycle.*

We define a binary relation θ on $V(G)$ by $x\theta y$ if $x = y$ or there is a walk that connects x and y. Clearly, θ is an equivalence relation on $V(G)$ and hence partitions

$V(K)$ into blocks $S_1,\ldots,S_t$. These blocks or the corresponding induced subgraphs (depending on the context) are called *connected components* of G. The number of connected components of G is denoted by $\omega(G)$. A graph G is *connected* if $\omega(G)=1$. An example of a connected graph and of a graph with four connected components are given in Fig. 2.7.

Lemma 2.11 *$S\subseteq V(G)$ is a connected component of G if and only if no proper superset $S'\supset S$ induces a connected subgraph of G.*

Theorem 2.12 *A graph G is connected if and only if $E(A,B)\neq\varnothing$ for every partition $\{A,B\}$ of $V(G)$.*

Proof. ($\Rightarrow$) Let G be a connected graph and $\{A,B\}$ a partition of $V(G)$. Take any $a\in A$ and $b\in B$. Now G is connected, so there is a path $x_1\ldots x_k$ that connects a and b. Since $x_1=a$ and $x_k=b$, there is a j such that $x_j\in A$ and $x_{j+1}\in B$ whence $E(A,B)\neq\varnothing$.

($\Leftarrow$) Suppose G is not connected and let S_1, …, S_ω be the connected components. Then Lemma 2.11 yields $E(S_1,\bigcup_{j=2}^{\omega}S_j)=\varnothing$. □

Theorem 2.13 *At least one of the graphs G, $\overline{G}$ is connected.*

Proof. Suppose that G is not connected and let S_1, …, S_ω, $\omega\geqslant 2$, be the connected components of G. Let us show that any pair of vertices in G is connected by a path. Take any $x,y\in V(G)$, $x\neq y$. If x and y belong to distinct connected components of G then $\{x,y\}\notin E(G)$ and hence $\{x,y\}\in E(\overline{G})$, so they are connected by an edge. If, however, x and y belong to the same connected component of G, say S_i, take any $j\neq i$ and any $z\in S_j$. Then x and z are connected by an edge in $\overline{G}$ and so are y and z. Therefore, $x\,z\,y$ is a path in $\overline{G}$ that connects x and y. □

We see from the proof of previous theorem that if G is not connected, then $\overline{G}$ is "very connected". We shall now introduce a numerical measure that enables us to express such statements formally.

The *distance* $d_G(x,y)$ between vertices x and y of a connected graph G is defined by $d_G(x,x)=0$, and in case $x\neq y$,

$$d_G(x,y)=\min\{k:\ \text{there is a path of length } k \text{ that connects } x \text{ and } y\}.$$

Theorem 2.14 *Let $G=(V,E)$ be a connected graph. Then (V,d_G) is a metric space, i.e. for all $x,y,z\in V$ the following holds:*

(D1) $d_G(x,y)\geqslant 0$;

(D2) $d_G(x,y) = 0$ *if and only if* $x = y$*;*

(D3) $d_G(x,y) = d_G(y,x)$*; and*

(D4) $d_G(x,z) \leqslant d_G(x,y) + d_G(y,z)$*.*

If G is obvious, instead of $d_G(x,y)$ we simply write $d(x,y)$. The *diameter* $d(G)$ of a connected graph G is the maximum distance between two of its vertices:

$$d(G) = \max\{d(x,y) : x,y \in V(G)\}.$$

Example 2.15 (a) $d(G) = 1$ if and only if G is a complete graph.

(b) $d(P_n) = n-1$ and $d(C_n) = \lfloor \frac{n-1}{2} \rfloor$.

A graph G is *bipartite* if there is a partition $\{X,Y\}$ of $V(G)$ such that every edge in G has one end in X and the other in Y, i.e. $E(G) = E(X,Y)$. Therefore, X and Y are independent sets. A *complete bipartite graph* is a bipartite graph with partition $\{X,Y\}$ of vertices such that its edges are *all* pairs $\{x,y\}$ with $x \in X$ and $y \in Y$. If $|X| = p$ and $|Y| = q$, the complete bipartite graph with the partition $\{X,Y\}$ is denoted by $K_{p,q}$. A *star* with n vertices, denoted by S_n, is a complete bipartite graph $K_{1,n-1}$. A bipartite graph, a $K_{3,4}$ and a star S_{10} are depicted in Fig. 2.8.

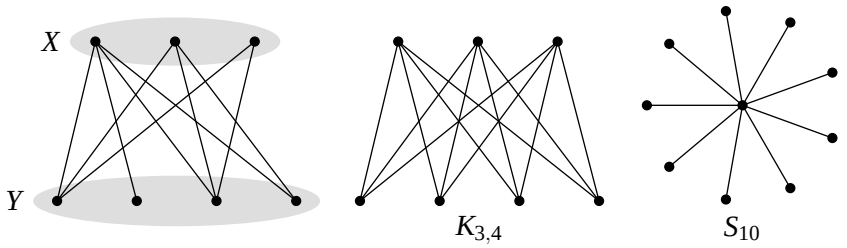


Figure 2.8: A bipartite graph, a $K_{3,4}$ and a star S_{10}

Lemma 2.16 *A graph G with at least two vertices is a bipartite graph if and only if every connected component of G is either an isolated vertex or a bipartite graph.*

Theorem 2.17 *A graph G with at least two vertices is bipartite if and only if G does not contain an odd cycle.*

Proof. According to Lemma 2.16 it suffices to give the proof for connected graphs. So, let G be a connected graph and $n(G) \geqslant 2$.

$(\Rightarrow)$ Let G be a bipartite graph and suppose G contains an odd cycle whose vertices are $v_1, v_2, \ldots, v_{2k+1}$. So v_i is adjacent to v_{i+1} for all $i \in \{1,\ldots,2k\}$ and v_{2k+1} is adjacent to v_1. Let $\{X,Y\}$ be a partition of $V(G)$ showing that G is bipartite,

i.e. such that $E(G[X]) = E(G[Y]) = \varnothing$. Now v_1 belongs to X or Y, so assume that $v_1 \in X$. Then $v_2 \in Y$ since v_2 is adjacent to v_1 and G is bipartite, and this forces $v_3 \in X$, $v_4 \in Y$ and so on. We see that vertices with odd indices belong to X, so $v_{2k+1} \in X$. But we have $x_1 \in X$ too, so $E(G[X])$ contains $\{x_1, x_{2k+1}\}$ which contradicts the assumption $E(G[X]) = \varnothing$.

($\Leftarrow$) Suppose G does not contain an odd cycle. Take any $v \in V(G)$ and define $A_0, A_1, \ldots \subseteq V(G)$ as follows:

$$A_n = \{x \in V(G) : d(v,x) = n\},$$

for $n \geqslant 0$. Since G is connected, there is a path connecting v to any other vertex of G, so each vertex of G appears in at least one of the A_i's. The A_i's are disjoint by the construction and the fact that $V(G)$ is finite now yields that there is an s such that $\{A_0, A_1, \ldots, A_s\}$ is a partition of $V(G)$ and $A_t = \varnothing$ for all $t > s$. Let

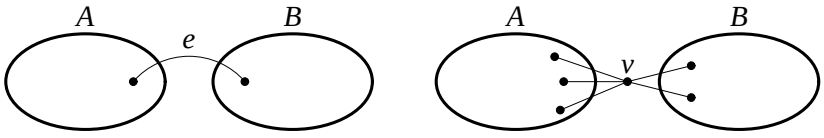
$$X = \bigcup_{j \text{ even}} A_j, \quad \text{and} \quad Y = \bigcup_{j \text{ odd}} A_j$$

and let us show that both X and Y are independent sets in G. Suppose that there are vertices $x, y \in X$ such that x and y are adjacent. By the construction of X there is an even path $v \ldots x$ and an even path $y \ldots v$. By chaining these two paths together with the edge $e = \{x, y\}$ we obtain a closed walk $v \ldots x\, e\, y \ldots v$ of odd length, so by Lemma 2.10 G contains an odd cycle, which is impossible.

The proof that Y is independent is analogous. Therefore, both X and Y are sets of independent vertices. This shows that G is a bipartite graph and one possible partition of its vertices is $\{X, Y\}$. □

Note that this theorem does not imply that bipartite graphs have to have cycles. A graph with no cycles is a bipartite graph, and this follows from the theorem since it has *no odd cycles*.

Let e be an edge and v a vertex of a graph G. By $G - e$ we denote the graph obtained from G by removing the edge e, while $G - v$ denotes the graph obtained from G by removing v and all the edges of G incident to v. A *cut-vertex* of a graph G is a vertex $v \in V(G)$ such that $\omega(G - v) > \omega(G)$. A *cut-edge* of a graph G is an edge $e \in E(G)$ such that $\omega(G - e) > \omega(G)$. Cut-vertices and cut-edges are weak points in the graph since removing one of these makes the graph split. Intuitively, they look like this:



a cut-edge a cut-vertex

Theorem 2.18 *Let e be an edge of a graph G. The following are equivalent:*

(1) e is a cut-edge of G;

(2) there is a partition $\{A,B\}$ of $V(G)$ such that $E(A,B)=\{e\}$;

(3) e belongs to no cycle of G.

Proof. We give the proof in case G is connected. If G is not connected it suffices to consider the connected component of G that contains e.

$(2)\Rightarrow(1)$: If $E(A,B)=\{e\}$ in G then $E(A,B)=\varnothing$ in $G-e$, so $G-e$ is not connected by Theorem 2.12. Therefore, $\omega(G-e)>1=\omega(G)$.

$(1)\Rightarrow(3)$: Suppose that e appears in a cycle

$$C = v_0 \underset{\uparrow}{e} v_1\ e_2\ v_2\ \ldots\ v_{k-1}\ e_k\ v_0$$

of G. To show that $G-e$ is connected take an arbitrary pair of vertices $x\neq y$. Since G is connected, there is a path P that connects x to y. If P does not contain e, it is also a path in $G-e$ that connects x to y. If, however, P contains e, say $P=x\ldots v_0\ e\ v_1\ldots y$, then remove e from P and replace it with $C-e$ to obtain the following walk:

$$W = x\ldots \underbrace{v_0\ e_k\ v_{k-1}\ldots v_2\ e_2\ v_1}_{C-e}\ldots y,$$

Fig. 2.9. Since e appears once in P and once in C it follows that e does not appear in W, so W is a walk from x to y in $G-e$.

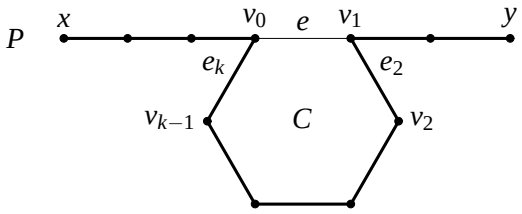


Figure 2.9: The walk W

$(3)\Rightarrow(2)$: Suppose that $e=\{a,b\}$ belongs to no cycle of G and define A and B as follows: $A=\{a\}\cup\{x\in V(G)$: there is a path from a to x that does not pass through $e\}$ and $B=V(G)\setminus A$. If $b\notin B$ then $b\in A$ and there is a path from a to b that does not pass through e. This path together with e forms a cycle that contains e. Since there are no such cycles we have $b\in B$. So, $\{A,B\}$ is a partition of $V(G)$ and $e\in E(A,B)$. Suppose now that there is an $e'\in E(A,B)$, $e'\neq e$, and let $e'=\{a',b'\}$, $a'\in A$, $b'\in B$, Fig. 2.10. We will assume further that $a\neq a'$ and $b\neq b'$ since these

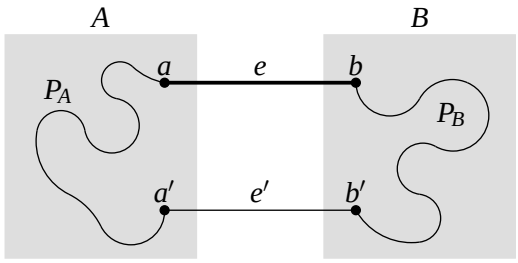


Figure 2.10: A cycle that contains e

two cases follow by similar arguments. There is a path $P_A = a \ldots a'$ that does not pass through e and there is a path $P_B = b' \ldots b$ that does not pass through e. Now these two paths together with e and e' form a cycle $\underbrace{a \ldots a'}_{P_A}\ e'\ \underbrace{b' \ldots b}_{P_B}\ e\ a$ which contains e. This contradiction shows that $E(A,B) = \{e\}$. □

Theorem 2.19 *Let v be a vertex of G. Then v is a cut-vertex of G if and only if there is a partition $\{A,B\}$ of $V(G) \setminus \{v\}$ such that $E(A,B) = \varnothing$, $E(A,\{v\}) \neq \varnothing$ and $E(B,\{v\}) \neq \varnothing$.*

Theorem 2.20 *If e is a cut-edge of G then $\omega(G-e) = \omega(G)+1$. If v is a cut-vertex of G then $\omega(G-v) < \omega(G)+\delta(v)$.*

Theorem 2.21 *If G is a connected graph with at least three vertices and if G has a cut-edge, then G has a cut-vertex.*

Proof. Let e be a cut-edge of G. Then there is a partition $\{A,B\}$ of $V(G)$ such that $E(A,B) = \{e\}$ (Theorem 2.18). Let $e = \{a,b\}$ and let $a \in A$ and $b \in B$. From $n(G) \geqslant 3$ it follows that $|A| \geqslant 2$ or $|B| \geqslant 2$, say $|A| \geqslant 2$. Since the graph is connected, a has a neighbour c in A, Fig. 2.11. Now let $A' = A \setminus \{a\}$ and note that

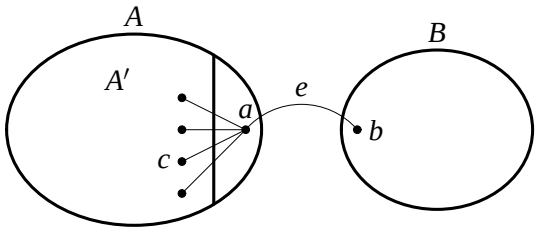


Figure 2.11: The proof of Theorem 2.21

$E(A',B) = \varnothing$, $E(A',\{a\}) \neq \varnothing$ and $E(B,\{a\}) \neq \varnothing$. Therefore, a is a cut-vertex according to Theorem 2.19. □

We have seen in Theorem 2.18 that a graph has no cut-edges if and only if every edge belongs to a cycle. The analogous statement for cut-vertices is the famous Whitney Theorem.

Theorem 2.22 (Whitney 1932) *Let G be a connected graph with at least three vertices. G has no cut-vertices if and only if any two vertices lie on a common cycle.*

Proof. ($\Leftarrow$) Since any two vertices u and v lie on a common cycle, removing one vertex from the graph cannot separate u from v, and hence $G-x$ is connected for all x.

($\Rightarrow$) For the converse, suppose that G has no cut-vertices. We say that two paths $ux_1 \ldots x_k v$ and $uy_1 \ldots y_l v$ connecting u to v are internally disjoint if $\{x_1,\ldots,x_k\} \cap \{y_1,\ldots,y_l\} = \varnothing$. Now take any u and v in G, $u \neq v$, and let us show by induction on $d(u,v)$ that G has two intenally disjoint paths connecting u and v. Clearly, the two paths will then form a cycle containing both u and v.

Let $d(u,v) = 1$ and let $e = \{u,v\}$. The graph $G-e$ is connected by Theorem 2.21 so there is a path in $G-e$ from u to v. This is also a path in G and it is internally disjoint from the trivial path $u\,v$ consisting of the edge e itself.

For the induction step, let $d(u,v) = k > 1$ and assume that G has internally disjoint paths connecting every pair of vertices x, y such that $1 \leqslant d(x,y) < k$. Let $u\,x_1\,\ldots\,x_{k-1}\,v$ the a path of length k (i.e. one of the shortest paths that connect u to v). We have $d(u,x_{k-1}) = k-1$, and hence by the induction hypothesis G has internally disjoint paths P and Q joining u to x_{k-1}, Fig. 2.12. Since $G-x_{k-1}$ is

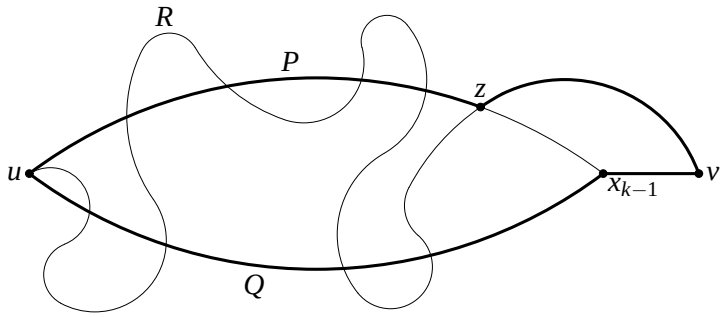


Figure 2.12: The proof of Whitney's Theorem

connected, $G-x_{k-1}$ contains a path R that joins u and v. If this path is internally

disjoint from P or Q we are done, so assume that R shares internal vertices with both P and Q. Let z be the last vertex of R belonging to $P \cup Q$. Without loss of generality we may assume that $z \in P$. We now combine the subpath of P joining u to z with the subpath of R joining z to v to obtain a path from u to v internally disjoint from the path $Q' = Q\,e'\,v$ where $e' = \{x_{k-1}, v\}$. □

2.3 Trees

A *tree* is a connected graph with no cycles. By Theorem 2.18 we see that every edge of a tree is a cut-edge. Therefore, a tree is a minimal connected graph with the given set of vertices. The following theorem shows that in a way trees capture the essence of the property of being connected.

Recall that a spanning subgraph of a graph $G = (V, E)$ is a graph $H = (W, E')$ such that $W = V$ and $E' \subseteq E$. If H is a tree, we say that H is a *spanning tree of* G.

Theorem 2.23 *A graph is connected if and only if it has a spanning tree.*

Proof. Clearly, if a graph G contains a connected spanning subraph H then G is also connected. Therefore if a graph has a spanning tree, it is connected. For the converse, take any connected graph G and construct a sequence of graphs G_0, G_1, G_2, ... as follows: $G_0 = G$; if G_i has a cycle, take any edge e_i that lies on a cycle and let $G_{i+1} = G_i - e_i$, otherwise put $G_{i+1} = G_i$. Each G_i is a spanning subgraph of G and each G_i is connected since an edge that lies on a cycle cannot be a cut-edge (Theorem 2.18). Moreover, if $G_i = G_{i+1}$ then $G_i = G_j$ for all $j > i$. Let m be the number of edges of G. Since we cannot remove more than m edges from G, we conclude that $G_{m+1} = G_{m+2}$. By construction of the sequence this means that G_{m+1} has no cycles. Therefore, G_{m+1} is a spanning tree of G. □

We will now show that each tree with n vertices has $n-1$ edges and that each two of the three properties listed below implies the remaining one:

- being connected,
- having no cycles, and
- $m = n - 1$.

Lemma 2.24 *Each tree with at least two vertices has at least two leaves.*

Proof. Let G be a tree with $n \geqslant 2$ vertices and let $v_1, v_2, \ldots, v_k$ be the longest path in the tree. Then $k \geqslant 2$ since G is a connected graph with at least two vertices. If $\delta(v_1) > 1$ then v_1 has a neighbour x distinct from v_2. If x is a new vertex, i.e.

$x \notin \{v_3, \ldots, v_k\}$, then the path x, v_1, v_2, ..., v_k is longer than the longest path in G, which is impossible. If, however, $x \in \{v_3, \ldots, v_k\}$ then G has a cycle, which contradicts the assumption that G is a tree. Therefore, v_1 is a leaf. The same argument shows that v_k is another leaf. □

Theorem 2.25 *Let $G = (V,E)$ be a tree with n vertices and m edges. Then $m = n-1$, and consequently $\sum_{v \in V} \delta(v) = 2(n-1)$.*

Proof. The second part of the theorem follows from the First Theorem of Graph Theory, so let us show that $m = n-1$. The proof is by induction on n. The cases $n = 1$ and $n = 2$ are trivial. Assume that the statement is true for all trees with less then n vertices and consider a tree G with n vertices. By Lemma 2.24 there is a leaf x in G. According Theorem 2.19 the degree of a cut-vertex is at least two, so x is not a cut-vertex and hence $G-x$ is connected. Clearly, $G-x$ does not have cycles (removing vertices and edges cannot introduce cycles), so $G-x$ is a tree with less than n vertices. By the induction hypothesis, $m' = n'-1$, where $m' = m(G-x)$ and $n' = n(G-x)$. But $m' = m-1$ and $n' = n-1$ since x is a leaf, whence $m = n-1$. □

Theorem 2.26 *Let G be a graph with n vertices and m edges. If $m = n-1$ and G has no cycles then G is connected (hence a tree).*

Proof. Suppose that $m = n-1$ and that G has no cycles. Let S_1, ..., S_ω be the connected components of G. Each connected component is a tree, so $m_i = n_i - 1$ for all i, where $m_i = m(S_i)$ and $n_i = n(S_i)$. Therefore $\sum_{i=1}^{\omega} m_i = \sum_{i=1}^{\omega} n_i - \omega$ i.e. $m = n - \omega$ (since $m = \sum_{i=1}^{\omega} m_i$ and $n = \sum_{i=1}^{\omega} n_i$). Now, $m = n-1$ yields $\omega = 1$, i.e. G is connected. □

Theorem 2.27 *Let G be a connected graph with $n \geqslant 2$ vertices and m edges and let $m = n-1$. Then G has no cycles (and hence it is a tree).*

Proof. According to Theorem 2.23 the graph $G = (V,E)$ has a spanning tree $H = (V,E')$. Since H is a tree Theorem 2.25 yields $m(H) = n(H) - 1 = n-1$. Assumption $m = n-1$ now implies $m(H) = m$ and thus from $E' \subseteq E$ we conclude $E' = E$. Therefore, $G = H$ and so G is a tree. □

Corollary 2.28 *A connected graph with n vertices and m edges is a tree if and only if $m = n-1$.*

We shall conclude the section by a result on the number of distinct trees. Let us first note that when counting structures we can count distinct structures and non-isomorphic structures. For example, there are 16 distinct trees on a four element set, but only two nonisomorphic, see Fig. 2.13. It is not surprising that counting nonisomorphic structures is more difficult.

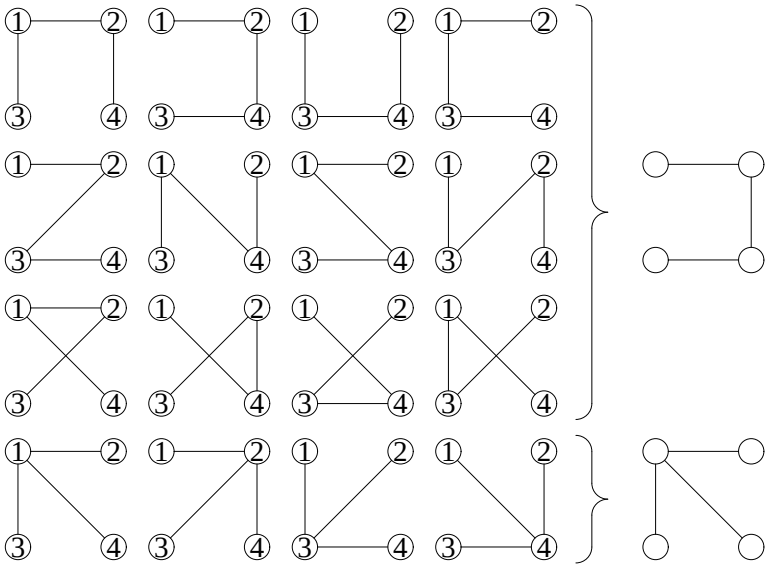


Figure 2.13: Sixteen distinct and only two nonisomorphic trees with four vertices

Theorem 2.29 (Cayley 1889) *There are n^{n-2} distinct trees with n vertices.*

Proof. Let $V = \{1,\dots,n\}$ be a finite set that serves as a set of vertices. The proof we are going to present is due to H. Prüfer[1]. The idea is to encode each tree on V by a sequence of integers $(a_1,\dots,a_{n-2})$ and thus provide a bijection $\varphi : \mathscr{T}_n \to \{1,2,\dots,n\}^{n-2}$, where $\mathscr{T}_n$ denotes the set of all trees on V.

We first show how to construct the Prüfer code of a tree. Let T be a tree with the set of vertices V. We shall construct a sequence of trees (T_i) and two sequences of integers, the code (a_i) and an auxiliary sequence (b_i). Let $T_1 = T$. Given T_i, let b_i be the smallest leaf of the tree (vertices are integers, so out of all integers that appear as leaves we choose the smallest) and let a_i be its only neighbour. Now put $T_{i+1} = T_i - b_i$ and repeat until a tree with two vertices is obtained. The code of the tree is now $(a_1,a_2,\dots,a_{n-2})$. An example is given in Fig. 2.14. Thus, we have a function $\varphi : \mathscr{T}_n \to \{1,\dots,n\}^{n-2}$ that takes a tree onto its Prüfer code.

Conversely, given a sequence $(a_1,\dots,a_{n-2})$ we can construct the tree as follows. For $S \subseteq \{1,\dots,n\}$ let $\operatorname{mix} S = \min(\{1,\dots,n\} \setminus S)$ denote the minimal number not in S (*mi*nimal ex*c*luded). Put $a_{n-1} = n$ and then construct b_1, b_2, …, b_{n-1} by

$$b_i = \operatorname{mix}\{a_i,\dots,a_{n-1},b_1,\dots,b_{i-1}\}$$

[1]H. Prüfer, *Neuer Beweis eines Satzes über Permutationen*, Archiv der Math. und Phys. (3) 27(1918), 142–144

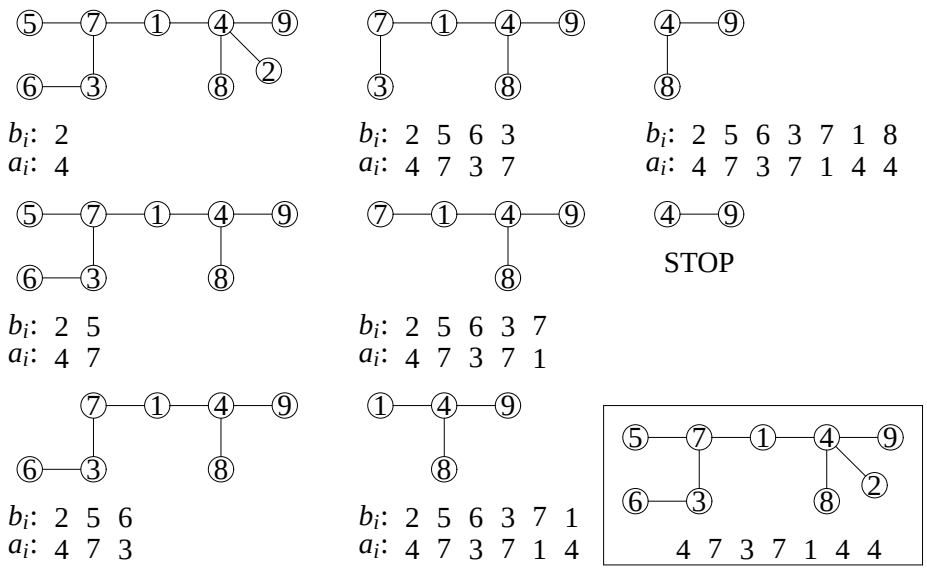


Figure 2.14: The Prüfer code of a tree

(for $i = 1$ there are no b_j's in the set). For example in case of $(4,7,3,4,1,4,4)$ we have $a_8 = 9$ and:

$$\begin{aligned}
b_1 &= \text{mix}\{4,7,3,4,1,4,4,9\} = 2\\
b_2 &= \text{mix}\{\ \ 7,3,4,1,4,4,9,2\} = 5\\
b_3 &= \text{mix}\{\ \ \ \ 3,4,1,4,4,9,2,5\} = 6\\
b_4 &= \text{mix}\{\ \ \ \ \ \ 4,1,4,4,9,2,5,6\} = 3\\
b_5 &= \text{mix}\{\ \ \ \ \ \ \ \ 1,4,4,9,2,5,6,3\} = 7\\
b_6 &= \text{mix}\{\ \ \ \ \ \ \ \ \ \ 4,4,9,2,5,6,3,7\} = 1\\
b_7 &= \text{mix}\{\ \ \ \ \ \ \ \ \ \ \ \ 4,9,2,5,6,3,7,1\} = 8\\
b_8 &= \text{mix}\{\ \ \ \ \ \ \ \ \ \ \ \ \ \ 9,2,5,6,3,7,1,8\} = 4
\end{aligned}$$

This process is called the *reconstruction procedure* since, as we shall see, it produces a tree whose Prüfer code is $(a_1,\ldots,a_{n-2})$.

Let us show that $\{\{b_i,a_i\} : 1 \leqslant i \leqslant n\}$ is the set of edges of a tree. If $i < j$ then, by construction, $b_j = \text{mix}\{a_j,\ldots,a_{n-1},b_1,\ldots,b_i,\ldots,b_{j-1}\}$, so $b_j \neq b_i$. We see that all b_i's are distinct and smaller than $n = a_{n-1}$. Therefore, $\{b_1,\ldots,b_{n-1}\} = \{1,\ldots,n-1\}$ and hence $\{b_1,\ldots,b_{n-1},a_{n-1}\} = \{1,\ldots,n-1,n\}$. Moreover, if $i \leqslant j$ then $a_j \notin \{b_1,\ldots,b_j\}$ since $b_i = \text{mix}\{a_i,\ldots,a_j,\ldots,a_{n-1},b_1,\ldots,b_{i-1}\}$, so from $\{b_1,\ldots,b_{n-1},a_{n-1}\} = \{1,\ldots,n-1,n\}$ it follows that $a_j \in \{b_{j+1},\ldots,b_{n-1},a_{n-1}\}$.

To summarize,

$$\begin{array}{l} a_j \in \{b_{j+1}, b_{j+2}, \ldots, b_{n-1}, a_{n-1}\} \text{ and} \\ b_j \notin \{a_{j+1}, b_{j+1}, a_{j+2}, b_{j+2}, \ldots, a_{n-1}, b_{n-1}\}, \end{array} \quad \text{for all } j. \qquad (\star)$$

To build the graph we start from $\{b_{n-1}, a_{n-1}\}$ and then add edges $\{b_{n-2}, a_{n-2}\}$, $\{b_{n-3}, a_{n-3}\}$, …, $\{b_1, a_1\}$ one by one. From $(\star)$ it follows that at each step we extend the graph by one new vertex b_i and one new edge $\{b_i, a_i\}$ that connects the new vertex to an existing one. Therefore, the graph we obtain at the end is connected, and a connected graph with n vertices and $n-1$ edges has to be a tree (Corollary 2.28). Thus, we have a function $\psi : \{1, \ldots, n\}^{n-2} \to \mathscr{T}_n$ that takes a code and produces a tree.

To complete the proof, we have to show that φ and ψ are inverses of one another, i.e. $\varphi \circ \psi = \mathrm{id}$ and $\psi \circ \varphi = \mathrm{id}$. We show only $\psi \circ \varphi = \mathrm{id}$ i.e. $\psi(\varphi(T)) = T$ for all $T \in \mathscr{T}_n$ (the other equality is left for Homework 2.10). For a tree T, a vertex $v \in V(T)$ is an *internal vertex* T if $\delta_T(v) > 1$. Let $\mathrm{int}(T)$ denote the set of all internal vertices of T.

Take any $T \in \mathscr{T}_n$, let $(a_1, \ldots, a_{n-2})$ be its Prüfer code and $(b_1, \ldots, b_{n-2})$ the auxiliary sequence. At the end of the procedure of constructing the Prüfer code two vertices remain the the graph, the vertex $a_{n-1} = n$ and its neighbour whom we denote by b_{n-1}. Starting from $(a_1, \ldots, a_{n-1})$ the reconstruction procedure produces a sequence of integers b'_1, …, b'_{n-1}. We will show that $b_i = b'_i$ for all i. Assume also that $n \geqslant 3$.

Since b_1 is adjacent to a_1 in T and $n \geqslant 3$, a_1 cannot be a leaf of T so $a_1 \in \mathrm{int}(T)$. The same argument shows that $a_2 \in \mathrm{int}(T - b_1)$, $a_3 \in \mathrm{int}(T - b_1 - b_2)$, and in general, $a_{i+1} \in \mathrm{int}(T - b_1 - \ldots - b_i)$. Since $\mathrm{int}(T - v) \subseteq \mathrm{int}(T)$ whenever v is a leaf of T and $n(T) \geqslant 2$, it follows that $\mathrm{int}(T - b_1 - \ldots - b_i) = \{a_{i+1}, \ldots, a_{n-2}\}$. In particular, $\mathrm{int}(T) = \{a_1, \ldots, a_{n-2}\}$. Since each vertex of a tree with at least two vertices is either a leaf or an internal vertex we obtain that

$$V(T - b_1 - \ldots - b_i) \setminus \mathrm{int}(T - b_1 - \ldots - b_i)$$

is the set of leaves of $T - b_1 - \ldots - b_i$. Now $V(T - b_1 - \ldots - b_i) = \{1, \ldots, n\} \setminus \{b_1, \ldots, b_i\}$ and $\mathrm{int}(T - b_1 - \ldots - b_i) = \{a_{i+1}, \ldots, a_{n-2}\}$, so the set of leaves of $T - b_1 - \ldots - b_i$ is

$$\begin{aligned} \Big(\{1, \ldots, n\} \setminus \{b_1, \ldots, b_i\}\Big) \setminus \{a_{i+1}, \ldots, a_{n-2}\} = \\ = \{1, \ldots, n\} \setminus \{a_{i+1}, \ldots, a_{n-2}, b_1, \ldots, b_i\}. \end{aligned}$$

It is now easy to show that $b_i = b'_i$ by induction on i. As we have seen, b_1 is a leaf of T, so $b_1 \in \{1, \ldots, n\} \setminus \{a_1, \ldots, a_{n-2}\}$. But b_1 is the smallest such integer, whence

$b_1 = \min(\{1,\ldots,n\} \setminus \{a_1,\ldots,a_{n-2}\}) = \mathrm{mix}\{a_1,\ldots,a_{n-2}\} = b'_1$. Assume that $b_j = b'_j$ for all $j \in \{1,\ldots,i\}$ and consider b_{i+1}. It is the smallest leaf in $T - b_1 - \ldots - b_i$ so, with the help of induction hypothesis

$$\begin{aligned} b_{i+1} &= \min(\{1,\ldots,n\} \setminus \{a_{i+1},\ldots,a_{n-2},b_1,\ldots,b_i\}) \\ &= \mathrm{mix}\{a_{i+1},\ldots,a_{n-2},b_1,\ldots,b_i\} = \mathrm{mix}\{a_{i+1},\ldots,a_{n-2},b'_1,\ldots,b'_i\} = b'_{i+1} \end{aligned}$$

Therefore, $\{a_i,b_i\} = \{a_i,b'_i\}$ for all i and the tree produced by the reconstruction procedure is T, the tree we started with. □

2.4 Digraphs

A *digraph* is an ordered pair $D = (V,E)$ where V is a nonempty finite set and E is an arbitrary subset of V^2 such that $(x,x) \notin E$ for all $x \in V$. Elements of V are called *vertices* of D, while elements of E are called *edges* of D. We shall often write $V(D)$ and $E(D)$ to denote the set of vertices and the set of edges of D, and $n(D)$ and $m(D)$ to denote the number of vertices and the number of edges of D. Instead of $(x,y) \in E$ we often write $x \to y$ or $x \underset{D}{\to} y$. If $x \to y$ we say that x is a *predecessor* of y and y is a *successor* of x. We also say that the edge (x,y) goes out of the vertex x and into the vertex y. The number of edges that go out of v iz called the *out-degree* of v and will be denoted by $\delta^+_D(v)$. The number of edges that go into v is called the *in-degree* of v and will be denoted by $\delta^-_D(v)$. Further, let,

$$I_D(v) = \{x \in V : x \to v\}, \qquad O_D(v) = \{x \in V : v \to x\},$$

denote the set of predecessors and the set of successors of v. Clearly, $\delta^-_D(v) = |I_D(v)|$ and $\delta^+_D(v) = |O_D(v)|$. The *total degree* of a vertex v is $\delta_D(v) = \delta^-_D(v) + \delta^+_D(v)$. If D is clear from the context, we simply write $\delta^-(v)$, $\delta^+(v)$, $I(v)$, $O(v)$ and $\delta(v)$.

A *source* of a digraph D is a vertex $v \in V(D)$ such that $\delta^-(v) = 0$ and $\delta^+(v) > 0$. A *sink* of a digraph D is a vertex $v \in V(D)$ such that $\delta^-(v) > 0$ and $\delta^+(v) = 0$. A *back-edge* in a digraph D is an edge $(x,y) \in E(D)$ such that $(y,x) \in E(D)$. If D has no back-edges then $I(v) \cap O(v) = \varnothing$ for every $v \in V(D)$.

If v is a vertex and e an edge of a digraph D then $D-e$ denotes the digraph obtained from D by removing the edge e, while $D-v$ denotes the digraph obtained from D by removing v, the edges that go into v and the edges that go out of v.

Digraphs also have a very natural graphical representation. Vertices are represented as points in a plane, while an edge $x \to y$ is represented as a directed curve (usually an arrow) going from x to y. Fig. 2.15 (a) depicts a digraph with 10 vertices.

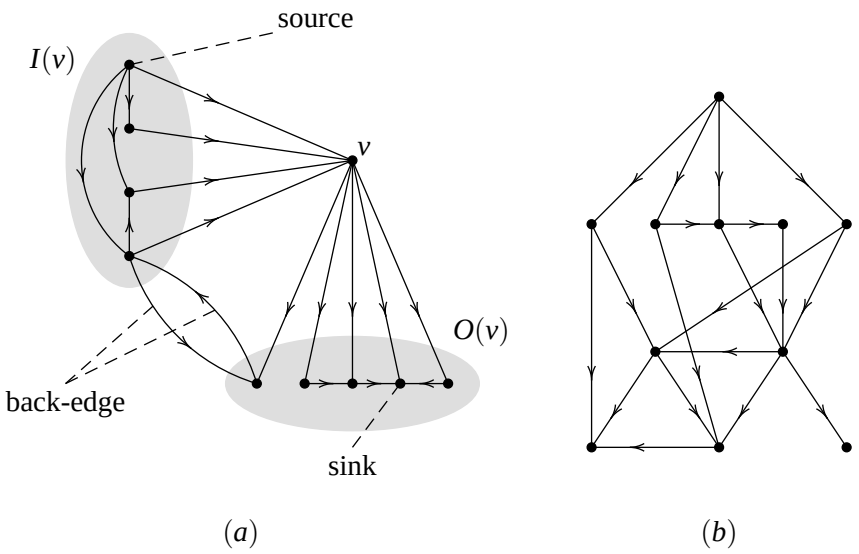


Figure 2.15: Two digraphs

Theorem 2.30 (The First Theorem for Digraphs) *Let $D = (V,E)$ be a digraph with m edges. Then $\sum_{v\in V} \delta^-(v) = \sum_{v\in V} \delta^+(v) = m$.*

Digraphs $D_1 = (V_1,E_1)$ and $D_2 = (V_2,E_2)$ are *isomorphic* if there exists a bijection $\varphi : V_1 \to V_2$ such that $(x,y) \in E_1$ if and only if $(\varphi(x),\varphi(y)) \in E_2$. The bijection φ is referred to as as *isomorphism* and we write $D_1 \cong D_2$.

The notions of the oriented path, oriented cycle and oriented walk in a digraph are straightforward generalizations of their "unoriented" versions. An *oriented walk* is a sequence of vertices and edges $x_0\ e_1\ x_1 \ldots x_{k-1}\ e_k\ x_k$ such that $e_i = (x_{i-1},x_i)$. We say that k is the *length of the walk*. An *oriented path* is an oriented walk where all vertices and all edges are distinct. An *oriented cycle* is an oriented walk where all edges and vertices are distinct, with the exception of $x_0 = x_k$.

Theorem 2.31 *Let D be a digraph with at least one edge. If D has no sinks, then it has an oriented cycle. Dually, if D has no sources, it has an oriented cycle.*

A digraph is *acyclic* if it has no oriented cycles. Fig. 2.15 (b) is an example of an acyclic digraph.

Corollary 2.32 *Each acyclic digraph with at least one edge has both a source and a sink.*

Theorem 2.33 *A digraph D with n vertices is acyclic if and only if it is possible to arrange its vertices as $(v_1,\ldots,v_n)$ in such a way that $v_i \to v_j$ implies $i<j$.*

Proof. ($\Leftarrow$) If such an arrangement of vertices exists then clearly G has no oriented cycles.

($\Rightarrow$) We use induction on n. Cases $n=1$ and $n=2$ are easy. Assume that such an arrangement of vertices exists for all acyclic digraphs with less than n vertices and let D be an acyclic digraph with n vertices. If there is a vertex x such that $\delta(v)=0$ put $v_1=x$. Otherwise, D has at least one edge, so it has a source. Let v_1 be any source of D. Now, $D-v_1$ is again an acyclic digraph and by induction hypothesis its vertices can be arranged into a sequence $(v_2,\ldots,v_n)$ in such a way that $v_i \to v_j$ implies $i<j$ for all $i,j \geqslant 2$. Since $I(v_1)=\varnothing$ and $O(v_1)\subseteq\{v_2,\ldots,v_n\}$, it is easy to see that $(v_1,v_2,\ldots,v_n)$ is the required arrangement of vertices of D. □

A digraph $D'=(V',E')$ is a *subdigraph* of a digraph $D=(V,E)$ if $V'\subseteq V$ and $E'\subseteq E$. We write $D'\leqslant D$. For $S\subseteq V$, the *subdigraph induced by S* is the digraph $D[S]=(S,S^2\cap E)$.

We say that $S\subseteq V(D)$ *dominates* D if $D[S]$ has no edges and the following holds: for every $x\in V(D)\setminus S$ there is an $s\in S$ such that either $s\to x$ or $s\to y\to x$ for some $y\in V(D)$.

Theorem 2.34 (Chvátal, Lovász 1974) *For every digraph D there is a set of vertices $S\subseteq V(D)$ which dominates D.*

Proof. We use induction on $n=n(D)$. For $n=1$ or $n=2$ the claim is obvious. Suppose the claim is true for all digraphs with less than n vertices and let D be a digraph with $n\geqslant 3$ vertices. Take any $x\in V(D)$ and let $A=V(D)\setminus(\{x\}\cup O(x))$. If $A=\varnothing$ then $S=\{x\}$ dominates D. If, however, $A\neq\varnothing$, by the induction hypothesis the digraph $D[A]$ has a set of vertices $S'\subseteq A$ that dominates $D[A]$. If there are no edges in $D[S'\cup\{x\}]$ then $S=S'\cup\{x\}$ dominates D. Otherwise, there is a $z\in S'$ such that $x\to_D z$ or $z\to_D x$. From $z\notin O(x)$ we conclude that $z\to_D x$, so $S=S'$ dominates D. □

There are two natural notions of connectedness for digraphs. It seems natural to be able to go from any vertex to any other vertex respecting the orientation of the edges, but sometimes we might wish to be able to do the same thing regardless of the orientation of edges.

A *base* of a digraph $D=(V,E)$ is a graph $G=(V,E')$ where $E'=\{\{x,y\}:(x,y)\in D\}$. A base of a digraph is obtained by replacing oriented edges of the digraph by nonoriented edges, see Fig. 2.16. A digraph D is *weakly connected* if its base is a connected graph. A digraph D is *strongly connected* if for every pair

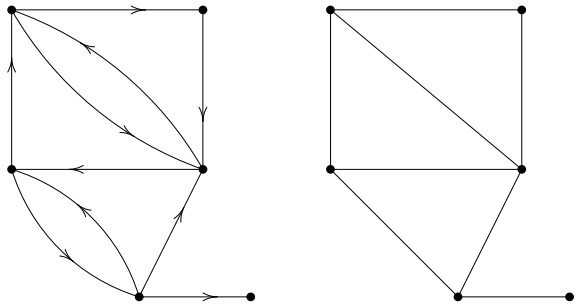


Figure 2.16: A digraph and its base

of vertices $x, y \in V$, $x \neq y$, there is an oriented path going from x to y, see Fig. 2.17.

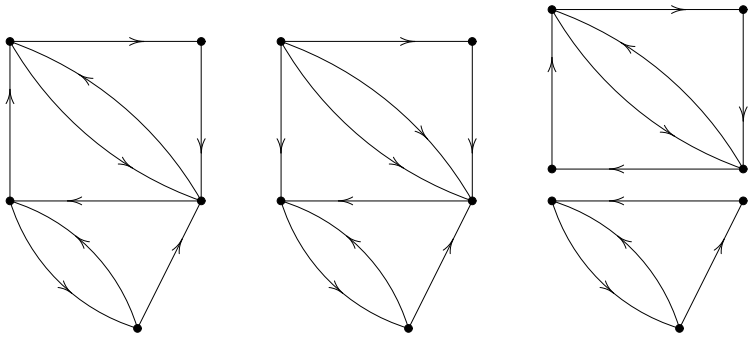


Figure 2.17: Two types of connectedness for digraphs

For disjoint $A, B \subseteq V(D)$ let $E(A,B) = \{(x,y) \in E(D) : x \in A, y \in B\}$ be the set of all edges of D that go from a vertex in A to a vertex in B.

Theorem 2.35 *A digraph D is weakly connected if and only if $E(A,B) \neq \varnothing$* ***or*** *$E(B,A) \neq \varnothing$ for every partition $\{A,B\}$ of $V(D)$.*

A digraph D is strongly connected if and only if $E(A,B) \neq \varnothing$ ***and*** *$E(B,A) \neq \varnothing$ for every partition $\{A,B\}$ of $V(D)$.*

Proof. We shall prove the second part of the theorem.

($\Rightarrow$) Let D be a strongly connected digraph and let $\{A,B\}$ be an arbitrary partition of $V(D)$. Take any $a \in A$ and any $b \in B$. The digraph D is strongly connected,

so there exists an oriented path from a to b. Since $a \in A$ and $b \in B$, the path has to cross from A into B at some point, so there exists an edge (x,y) along this path such that $x \in A$ and $y \in B$. Therefore, $E(A,B) \neq \varnothing$. Similarly, $E(B,A) \neq \varnothing$.

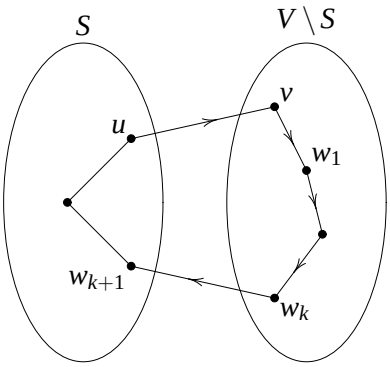
($\Leftarrow$) Take any $x,y \in V(D)$, $x \neq y$, and let us show that there is an oriented path from x to y. Let $A = \{x\} \cup \{v \in V(D) : \text{there is an oriented path from } x \text{ to } v\}$. We wish to show that $y \in A$. Suppose this is not the case and let $B = V(D) \setminus A$. Then $y \in B$ and so $B \neq \varnothing$. Now, $\{A,B\}$ is a partition of $V(D)$ and by the assumption $E(A,B) \neq \varnothing$. This means that there is a $v \in A$ and a $w \in B$ such that $v \to w$. But $v \in A$ means that there is an oriented path from x to v, so $v \to w$ implies that there is an oriented path from x to $w \notin A$. This contradiction shows that $y \in A$ and hence there is an oriented path from x to y. □

Every connected graph $G = (V,E)$ can be turned into a strongly connected digraph $D(G) = (V,E')$ where $E' = \{(x,y) : \{x,y\} \in E\}$, that is, by replacing each edge $\{x,y\}$ of G by a pair of edges (x,y), (y,x). Therefore, each connected graph is a base of some strongly connected digraph, possibly with back-edges. The following theorem shows that this is not the case if we forbid back-edges.

Theorem 2.36 *A connected graph G with at least two vertices is a base of a strongly connected digraph with no back-edges if and only if G has no cut-edges.*

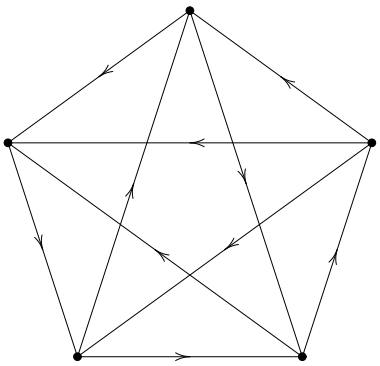
Proof. ($\Rightarrow$) Let $G = (V,E_G)$ be a base of a digraph $D = (V,E_D)$ and suppose that G has a cut-edge $e = \{u,v\}$. Then by Theorem 2.18 there is a partition $\{A,B\}$ of V such that $E_G(A,B) = \{e\}$. Since D has no back-edges then either $(u,v) \in E_D$ or $(v,u) \in E_D$, but not both. Therefore, either $E_D(A,B) = \varnothing$ or $E_D(B,A) = \varnothing$. In any case, D is not strongly connected by Theorem 2.35.

($\Leftarrow$) Let $G = (V,E)$ be a graph with no cut-edges and let $S \subseteq V$ be a maximal set of vertices such that $G[S]$ is a base of a strongly connected digraph $D(S)$ with no back-edges. Let us show that $S \neq \varnothing$. Note first that G contains a cycle (G has no cut-edges, so by Theorem 2.18 every edge of G belongs to a cycle; hence there is at least one cycle in G). Take any cycle C in G, orient its edges to obtain an oriented cycle and orient the remaining edges in $G[V(C)]$ arbitrarily. We thus obtain a strongly connected digraph $D(C)$ with no back-edges whose base is $G[V(C)]$. Therefore, there exists a set $S' \subseteq V$ with at least three vertices such that $G[S']$ is a base of a strongly connected digraph with no back-edges, so the maximal such set cannot be empty.



Let us show that $S = V$. Suppose to the contrary that $S \subset V$, i.e. $V \setminus S \neq \varnothing$. Since G is connected we have $E(S, V \setminus S) \neq \varnothing$, so take any $e = \{u, v\}$ such that $u \in S$ and $v \in V \setminus S$. There are no cut-edges in G so according to Theorem 2.18 the edge e belongs to a cycle in G. Let $v\, w_1 \ldots w_k$ be a part of the cycle that belongs to $V \setminus S$ and let w_{k+1} be the vertex that follows w_k on the cycle. By assumption, $w_{k+1} \in S$. Now orient the edges on the path $u\, v\, w_1 \ldots w_k\, w_{k+1}$ to obtain an oriented path that goes from u to w_{k+1} and attach the path to the digraph $D(S)$. Orient the remaining edges in $G[S \cup \{v, w_1, \ldots w_k\}]$ arbitrarily. The digraph D' obtained this way is strongly connected, has no back-edges and its base is $G[S \cup \{v, w_1, \ldots w_k\}]$ whose set of vertices is a proper superset of S. This contradiction shows that $S = V$, i.e. that G is a base of a strongly connected digraph with no back-edges. □

2.5 Tournaments



A *tournament* is a digraph $T = (V, E)$ with the property that for each pair $x, y \in V$, $x \neq y$, either $(x, y) \in T$ or $(y, x) \in T$. Equivalently, a tournament is a digraph with no back-edges whose base is a complete graph. Tournaments (as digraphs) appear as models of tournaments (as sprot events) where no match ends in a draw; each arrow then represents one match and goes from the vertex representing the winner to the vertex representing the loser.

A tournament with n vertices has $\binom{n}{2}$ edges and $\delta^+(v) + \delta^-(v) = n - 1$ for each vertex v. Therefore, it has become customary to consider only $\delta^+(v)$. When working with tournaments, $\delta^+(v)$ is called the *score* of v and denoted by $s(v)$. A tournament is *transitive* if $x \to y$ and $y \to z$ implies $x \to z$ whenever x, y and z are three distinct vertices of the tournament.

Theorem 2.37 *Let T be a tournament with n vertices. Then the following are equivalent:*

(1) T is an acyclic tournament;

(2) T is a transitive tournament;

(3) the scores of vertices in T are $0, 1, \ldots, n-1$.

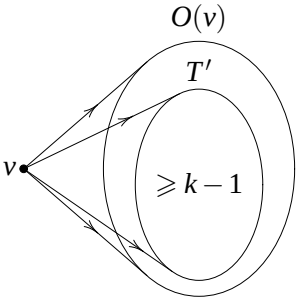
Proof. $(1) \Rightarrow (2)$: Suppose T is not a transitive tournament. Then there exist distinct vertices x, y and z such that $x \to y$ and $y \to z$ but $x \not\to z$. Since T is a tournament, $x \not\to z$ means that $z \to x$ and we obtain a cycle $x \to y \to z \to x$.

$(2) \Rightarrow (3)$: The proof is by induction on n. Cases $n = 2$ and $n = 3$ are trivial. Suppose that in each transitive tournament with $k < n$ vertices the scores of vertices are $0, 1, \ldots, k-1$ and let T be a transitive tournament with n vertices. Let v_1 be the vertex of T with maximal score and let us show that $s(v_1) = n-1$. Suppose that there is a vertex x such that $x \to v_1$. Then due to transitivity $v_1 \to z$ implies $x \to z$ and hence $s(x) \geqslant 1 + s(v_1) > s(v_1)$, which is impossible. Therefore, $v_1 \to x$ for all $x \neq v_1$ and hence $s(v_1) = n-1$. It is easy to see that $T - v_1$ is again a transitive tournament and by the induction hypothesis the scores of its vertices are $0, 1, \ldots, n-2$. Therefore, the scores of vertices in T are $0, 1, \ldots, n-2, n-1$.

$(3) \Rightarrow (1)$: The proof is again by induction on n and the cases $n = 2$ and $n = 3$ are trivial. Suppose that each tournament with $k < n$ vertices and with scores $0, 1, \ldots, k-1$ is acyclic and let T be a tournament with n vertices and scores $0, 1, \ldots, n-1$. Let v be the vertex of T whose score is $n-1$ and let C be an oriented cycle in T. Since $T - v$ is a tournament with scores $0, 1, \ldots, n-2$, it is acyclic according to the induction hypothesis so $V(C) \not\subseteq V(T-v)$. Therefore, C has to pass through v. On the other hand, $s(v) = n-1$ means that $v \to x$ for every $x \neq v$ so no cycle in T can pass through v. Contradiction. □

Corollary 2.38 *Two transitive tournaments are isomorphic if and only if they have the same number of vertices.*

Theorem 2.39 *Every tournament with at least 2^{k-1} vertices, $k \geqslant 2$, has a transitive subtournament with at least k vertices.*



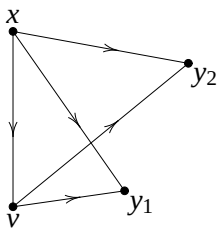
Proof. The proof is by induction on k. If $k = 2$ the tournament has at least two vertices and hence at least one edge, and each edge $x \to y$ is a transitive tournament with two vertices. Assume the claim is true for all integers less than k and consider a tournament T with at least 2^{k-1} vertices. Take any $v \in V(T)$. Then $V(T) = I(v) \cup \{v\} \cup O(v)$, so one of the sets $I(v)$, $O(v)$ has at least 2^{k-2} vertices. Without loss of generality we can assume that $|O(v)| \geqslant 2^{k-2}$. Inductioin hypothesis now yields that there is a transitive subtournament T' of $T[O(v)]$ with at least $k-1$ vertices. Then T' to-

gether with v induces a transitive subtournament of T with at least k vertices. □

A *king* in a tournament T is a vertex $v \in V(T)$ such that $\{v\}$ dominates T. This means that for every $x \neq v$ either $v \to x$ or $v \to y \to x$ for some $y \in V(T)$.

Theorem 2.40 *Each tournament with at least two vertices has a king.*

Proof. Let v be a vertex of T whose score is maximal and let us show that v is a king. Suppose to the contrary that v is not a king. Then there is an $x \neq v$ such that $v \not\to x$ and no $y \in V(T)$ satisfies $v \to y \to x$. Since T is a tournament, $v \not\to x$ means $x \to v$, while the other condition means that if $v \to y$ then $x \to y$. But then $s(x) \geqslant 1 + s(v) > s(v)$, which contradicts the maximality of $s(v)$. □



Homework

2.1. An *automorphsm* of a graph G is every isomorhism $\varphi : V(G) \to V(G)$ from the graph onto itself. By $\mathrm{Aut}(G)$ we denote the set of all the automorphisms of G.

(a) Show that $(\mathrm{Aut}(G), \circ)$ is a group.

(b) Describe $\mathrm{Aut}(K_n)$, $\mathrm{Aut}(S_n)$ and $\mathrm{Aut}(P_n)$ for $n \geqslant 3$.

(c) Show that $\mathrm{Aut}(G) = \mathrm{Aut}(\overline{G})$.

2.2. (a) Show that for every $n \geqslant 6$ there exists a graph G with n vertices such that $|\mathrm{Aut}(G)| = 1$.

(b) Show that for every $k \geqslant 2$ and every $n \geqslant k+3$ there exists a graph G with n vertices such that $|\mathrm{Aut}(G)| = k!$.

2.3. Prove Lemma 2.10.

2.4. Prove Theorem 2.14.

2.5. If G is not connected show that $d(\overline{G}) \leqslant 2$. (We know that $\overline{G}$ is connected).

2.6. Prove Theorem 2.19.

2.7. Prove Theorem 2.20.

2.8. Show that a graph is a tree if and only if each pair of distinct vertices of of the graph is connected by a unique path.

2.9. Find the number of distinct spanning trees of K_n.

2.10. Complete the proof of Theorem 2.29 by showing that $\varphi \circ \psi = \mathrm{id}$.

2.11. Prove Theorem 2.31.

2.12. In the distant land of Xÿç there are n cities some of which are connected by roads, but still it is possible to reach each city from every other city by traveling along the roads (and possibly passing through some other cities). The Evil Magician who rules the Xÿç would like to terrorize his people by making each road a one-way road in such a way that after leaving a city it is impossible to get back. Show that it is possible to do such a thing.

2.13. Prove the first part of Theorem 2.35 (the characterization of weak connectedness).

2.14. Prove Corollary 2.38.

2.15. A tournament is *regular* if $s(x) = s(y)$ for all x and y. Show that in a regular tournament each vertex is a king.

Exercises

2.16. Let G be a graph with n vertices and m edges. Show that $\Delta(G) \geqslant \frac{2m}{n}$.

2.17. Which of the following integer sequences can be a sequence of degrees of vertices of a graph?

(a) $(1,2,2,4,5,6,7)$;

(b) $(1,1,2,2,2,3,3)$;

(c) $(1,1,3,3,3,3,5,6,8,9)$.

†**2.18.** Show that there are

(a) $2^{\binom{n}{2}}$ distinct graphs with n vertices;

(b) $2^{\binom{n-1}{2}}$ distinct graphs with n vertices such that the degree of each vertex in the graph is even.

2.19. Let G be a graph with $\delta(G) \geqslant 2$. Then G contains a path of length $\geqslant \delta(G)$ and a cycle of length $\geqslant \delta(G)+1$.

2.20. Let G be a bipartite graph (not necessarily a complete bipartite graph!) with n vertices and m edges. Show that $m \leqslant \frac{1}{4}n^2$.

2.21. By $\alpha(G)$ we denote the maximum carinality of an independent set of vertices in G. Show that a graph G is bipartite if and only if every subgraph H of G satisfies $\alpha(H) \geqslant \frac{1}{2}n(H)$.

2.22. A *k-dimensional hypercube* is a graph $Q_k = (V_k, E_k)$ where V_k is the set of all 01-words of length k and $a_1 \dots a_k$, $b_1 \dots b_k \in V_k$ are adjacent if and only if the two words differ at exactly one place. For example, if $k = 4$ then 0101 and 0001 are adjacent in Q_4 while 0101 and 0000 are not.

(a) Find the number of vertices and the number of edges of Q_k.

(b) Show that Q_k is bipartite.

(c) Show that Q_k is connected and find $d(Q_k)$.

2.23. Show that for every even $n \geqslant 6$ there exists a connected regular graph of degree 3 with n vertices and with no trianlges.

2.24. Show that if $\delta(G) \geqslant \frac{1}{2}n(G)$ then G is connected and $d(G) \leqslant 2$.

2.25. Show that for every graph G there exists a regular graph H such that G is an induced subgraph of H and $\Delta(G) = \Delta(H)$.

2.26. Show that $\delta(\overline{G}) = (n(G)-1) - \Delta(G)$ and $\Delta(\overline{G}) = (n(G)-1) - \delta(G)$.

2.27. Show the following:

(a) If G is connected and $d(G) \geqslant 3$ then $\overline{G}$ is connected and $d(\overline{G}) \leqslant 3$.

(b) Every selfcomplementary graph G with at least two vertices is connected and $2 \leqslant d(G) \leqslant 3$.

2.28. Suppose that the degree of every vertex in a connected graph G is even. Show that $\omega(G-v) \leqslant \frac{1}{2}\delta(v)$ for all $v \in V(G)$.

2.29. Let $G = (V,E)$ be a connected graph with n vertices and let u be an arbitrary vertex of G. Show that $\sum_{x \in V} d(u,x) \leqslant \binom{n}{2}$.

2.30. Let G be a connected graph with at least two vertices. Show that G has at least two vertices that are not cut-vertices.

2.31. Show that if v is a cut-vertex of G, then v is not a cut-vertex of $\overline{G}$.

2.32. Show that each tree G has at least $\Delta(G)$ leaves.

2.33. Let T be a tree, $\Delta = \Delta(T)$ and f_k the number of vertices in T of degree k. Show that $f_1 = 2 + \sum_{k=3}^{\Delta} (k-2) f_k$.

2.34. Find all trees G such that $\overline{G}$ is a tree.

2.35. For every $n \geqslant 4$ find a graph G with n vertices such that for each $k \in \{2, \dots, n-2\}$ there is a spanning tree of G whose diameter is k.

2.36. Note first that each tree is a bipartite graph since no cycles means no odd cycles. Let $\{X,Y\}$ be a partition of the vertices of a tree T which demonstrates that T is a bipartite graph and assume that $|X| = |Y| + p$ for some $p > 0$. Show that X contains at least $p+1$ leaves of T.

2.37. A *forest* is a graph whose connected components are trees. Show that G is a forest if and only if $\delta(H) \leqslant 1$ for all induced subgraphs H of G.

†2.38. How many nonisomorphic spanning trees does $K_{2,n}$ have?

†2.39. Show that each spanning tree of a connected graph contains all cut-edges of the graph.

†2.40. A *block* of a connected graph G is a maximal set of vertices $S \subseteq V(G)$ such that $G[S]$ has no cut-vertices (that is, if $S' \supseteq S$ and $G[S']$ has no cut-vertices then $S' = S$).

(a) Show that any two blocks of a graph have at most one vertex in common.

(b) Let $B_1, \ldots, B_k$ be blocks of G and let $\mathscr{B}_G$ be the graph with vertices $\{1,\ldots,k\}$ where i is adjacent to j if and only if $i \neq j$ and B_i and B_j have a nonempty intersection. Show that $\mathscr{B}_G$ is a tree.

†2.41. Let $D = (V,E)$ be a weakly connected digraph. A strongly connected component of D is a maximal set of vertices $S \subseteq V$ such that $D[S]$ is strongly connected (that is, if $S' \supseteq S$ and $D[S']$ is strongly connected then $S' = S$).

(a) Show that $S \cap S' = \varnothing$ whenever S and S' are distinct strongly connected components of D.

(b) Let S and S' be distinct strongly connected components of D. Show that if $E(S,S') \neq \varnothing$ then $E(S',S) = \varnothing$.

(c) Let $S_1, \ldots, S_k$ be strongly connected components of D and let $\mathscr{S}_D$ be the graph with vertices $\{1,\ldots,k\}$ where $i \to j$ if and only if $i \neq j$ and $E(S_i,S_j) \neq \varnothing$. Show that $\mathscr{S}_D$ has no back-edges and its base is a tree.

2.42. Show that $\sum_{v \in V} (\delta^+(v))^2 = \sum_{v \in V} (\delta^-(v))^2$ in every tournament $T = (V,E)$.

2.43. A tournament is *regular* if $s(x) = s(y)$ for all x and y. Show that for each odd integer $n \geqslant 3$ there exists a regular tournament with n vertices.

2.44. Scores $s_1 \leqslant s_2 \leqslant \ldots \leqslant s_n$ of a tournament T satisfy $\sum_{i=1}^{k} s_i = \binom{k}{2}$ for every $k \in \{1,\ldots,n\}$. Show that T is an acyclic tournament.

Chapter 3

Eulerian and Hamiltonian graphs

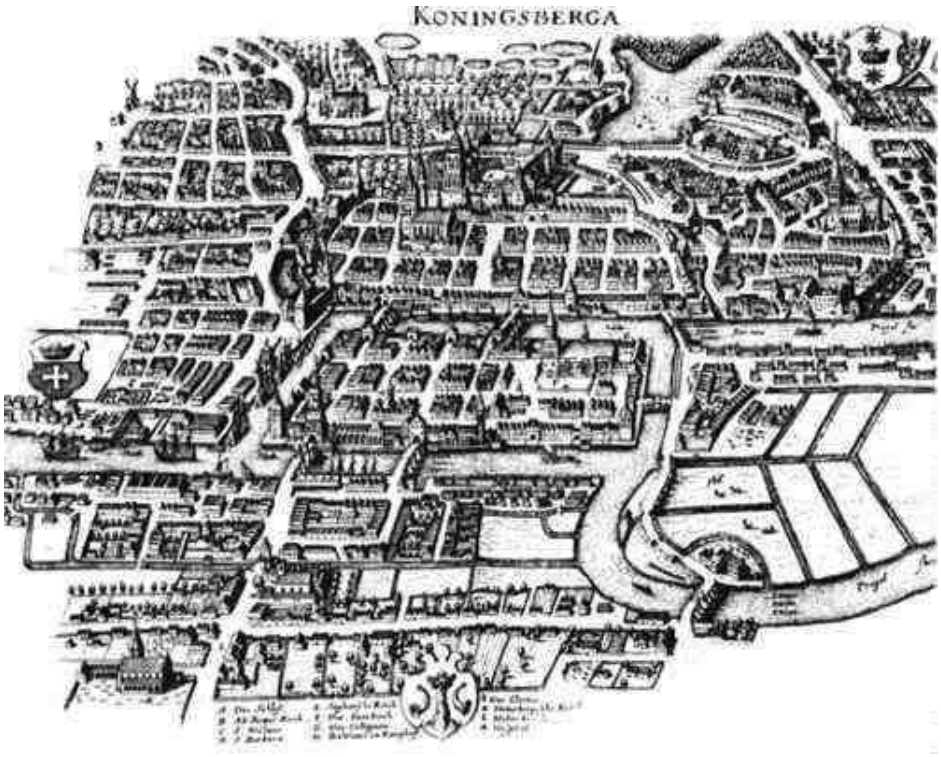
In this chapter we deal with two important classes of graphs:

- Eulerian graphs, which are graphs with the closed walk in which each edge occurs precisely once; and
- Hamiltonian graphs, which are graphs with the cycle in which every vertex occurs precisely once.

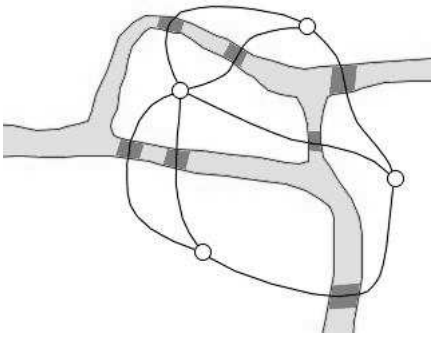
We present an easy characterisation of Eulerian graphs and discuss several necessary and sufficient conditions for a graph to be Hamiltonian. The fact that there is no "easy" and "useful" characterisation of Hamiltonian graphs is justified by the discussion at the end of the chapter where we argue that checking for a Hamiltonian cycle in a graph is an NP-complete problem.

3.1 Eulerian graphs

The famous Swiss mathematician Leonhard Euler was visiting the city of Königsberg in the year 1735. Königsberg was a city in Prussia situated on the Pregel River, which served as the residence of the dukes of Prussia in the 16th century. (Today, the city is named Kaliningrad, and is a major industrial and commercial center of western Russia.) The river Pregel flowed through the city such that in its center was an island, and after passing the island, the river broke into two parts. Seven bridges were built so that the people of the city could get from one part to another. A map of the center of Königsberg in 1735 looked like this:



A favorite pastime for visitors to the city was to try to cross each of the bridges of Königsberg exactly once. Euler was told by some people that it was impossible and by others that they doubted whether or not it could be done. No one believed it was possible. Eventually, Euler realized that all problems of this form could be represented by replacing areas of land by vertices, and the bridges to and from them by edges of a graph such as:



The problem now becomes to draw this picture without tracing any line twice and without picking the pencil up off the paper. All four of the vertices in the above picture have an odd degree. Take one of these vertices, say one of the ones of degree three. We could start at that vertex, and then arrive and leave later. But then we can't come back. So, every vertex with an odd degree has to be either the beginning or the end of the pencil-path and thus we can have at most two odd vertices. Therefore it is impossible to draw the above picture in one pencil stroke without tracing some line twice.

This is the first recorded problem in graph theory, and W. Tutte, himself a prominent graph-theorist, decided to celebrate the problem with a poem:

From Königsberg to König's book
by William T. Tutte

Some citizens of Koenigsberg
Were walking on the strand
Beside the river Pregel
With its seven bridges spanned.

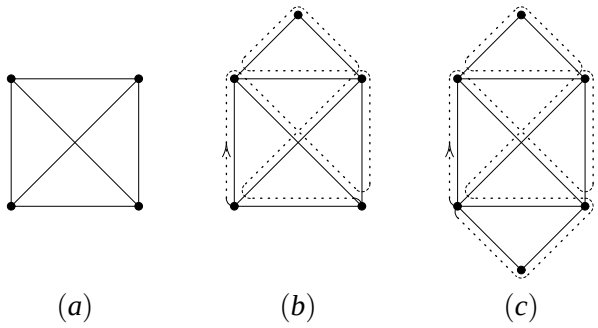
O, Euler, come and walk with us
Those burghers did beseech
We'll walk the seven bridges o'er
And pass but once by each.

"It can't be done" then Euler cried
"Here comes the Q.E.D.
Your islands are but vertices,
And all of odd degree."

We shall now go for a more formal treatment of this and similar problems. We shall first solve the general problem in case of oriented graphs, and then infer the solution in case of undirected graphs.

Definition 3.1 A *trail* in a graph is a walk in which edges are not allowed to repeat. An *Eulerian trail* in a graph is a trail that contains each edge of the graph precisely once. A graph is said to be *Eulerian* if it contains a closed Eulerian trail, Fig. 3.1.

Definition 3.2 Analogously, an *oriented trail* in a digraph is an oriented walk in which edges are not allowed to repeat. An *Eulerian trail* in a digraph is an oriented trail in the digraph that contains each edge of the digraph precisely once. A digraph is said to be *Eulerian* if it contains a closed Eulerian trail.

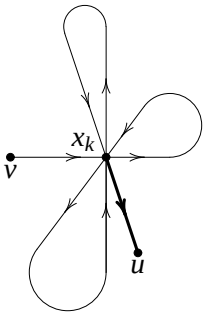


(a) (b) (c)

Figure 3.1: (a) A graph with no Eulerian trail; (b) a non-eulerian graph with an Eulerian trail; (c) an Eulerian graph

Lemma 3.3 *Let D be a digraph with no isolated vertices and with the property that $\delta^-(v) = \delta^+(v)$ for every $v \in V(D)$. Then every vertex of D belongs to a closed oriented trail in D.*

Proof. Let $W = v\, e_1\, x_1 \ldots e_k\, x_k$ be the longest trail in D that starts with v and let us show that $x_k = v$. Suppose to the contrary that $x_k \neq v$ and assume that x_k appears $l \geqslant 1$ times on the trail W. Each appearance of x_k on W engages one edge that leads into x_k and one edge that leads out of x_k, except for the last appearance of x_k that engages one edge leading into x_k. Therefore, W contains l edges leading into x_k and $l-1$ edges leading out of x_k. Since $\delta^-(x_k) = \delta^+(x_k)$, there exists an edge $e' = (x_k, u) \in E(D)$ that does not appear in W. Now, $v\, e_1\, x_1 \ldots e_k\, x_k\, e'\, u$ is a trail that starts from v longer that W. Contradiction. □



Theorem 3.4 *Let D be a digraph with no isolated vertices. Then D is an Eulerian digraph if and only if D is weakly connected and $\delta^-(v) = \delta^+(v)$ for every $v \in V(D)$.*

Proof. $(\Rightarrow)$ Let D be an Eulerian digraph with no isolated vertices and consider a closed Eulerian trail W in D. Walking along W we can start from any vertex in D and reach any other vertex in D which shows that D is strongly, and hence also weakly connected. The trail W can be partitioned into oriented cycles C_1, $\ldots$, C_k in such a way that every edge in D belongs to exactly one of the cycles (Homework 3.1). Each vertex of D appears on W, so each vertex belongs to at least one of the cycles. Now, if $v \in V(D)$ lies on exactly l of these cycles, then $\delta^-(v) = l = \delta^+(v)$ since every edge in W belongs to precisely one of the cycles

$C_1, \ldots, C_k$, and each of the cycles "absorbs" one edge that goes into v and one edge that goes out of v.

($\Leftarrow$) Take any $v \in V(D)$. According to Lemma 3.3, v belongs to some closed oriented trail in D. Let W be the longest closed oriented trail in D that contains v and let us show that W is an Eulerian trail in D.

Suppose that W is not an Eulerian trail in D, i.e. $E(W) \subset E(D)$. If $V(W) = V(D)$, take any $e = (u,v) \in E(D) \setminus E(W)$. If $V(W) \subset V(D)$ then $\{V(W), V(D) \setminus V(W)\}$ is a partition of $V(D)$ and since D is weakly connected there is an edge $e = (u,v) \in E(D) \setminus E(W)$ such that $u \in V(W)$ and $v \in V(D) \setminus V(W)$ (or the other way around; the proof is analogous). In any case, let S be the weak connected component of $D - E(W)$ that contains e. Since W is a closed trail, it is easy to see that $\delta_S^-(v) = \delta_S^+(v)$ for every $v \in V(S)$. Hence, by Lemma 3.3 there exists a closed trail W' in S that contains u. Since $E(W') \subseteq E(S) \subseteq E(D) \setminus E(W)$, it follows that $E(W') \cap E(W) = \varnothing$, so glueing W and W' at u provides a trail that contains v and which is longer than W. Contradiction. □

The characterisation of Eulerian graphs is similar, and the proof goes along the same guidelines as in case of digraphs.

Theorem 3.5 *Let G be a graph with no isolated vertices. Then G is an Eulerian graph if and only if G is connected and each vertex of G is even.*

Proof. Analogous to the proof of Theorem 3.4. □

It is now easy to characterize noneulerian graphs that contain an Eulerian trail (which therefore cannot be a closed Eulerian trail).

Theorem 3.6 *Let G be a noneulerian graph with no isolated vertices. Then G has an Eulerian trail if and only if it is connected and has precisely two odd vertices.*

Proof. ($\Rightarrow$) Let W be an Eulerian trail in G. Since G is not Eulerian, W is not closed. Denote the vertices it starts and ends with by u and v. Introduce a new vertex $x \notin V(G)$ and two new edges $\{x,u\}$, $\{x,v\}$, and apply Theorem 3.5.

($\Leftarrow$) Let u and v be the odd vertices in G. Introduce a new vertex $x \notin V(G)$ and two new edges $\{x,u\}$, $\{x,v\}$, and apply Theorem 3.5. □

Finally, we conclude the section with another characterization of Eulerian graphs.

Theorem 3.7 *Let G be a connected graph. Then G is Eulerian if and only if every edge of G belongs to an odd number of cycles in G.*

Proof. We start by proving an auxiliary statement.

Claim. Let G be a connected noneulerian graph with an Eulerian trail and let u and v be the only two odd vertices in G. Then the number of trails that start at u, end in v and where v appears only once (i.e. at the end of the trail) is odd.

Proof. The proof is by induction on $m(G)$. The claim is true for connected noneulerian graphs with an Eulerian trail that have 1, 2 and 3 edges. Suppose the claim holds for all such graphs with $< m$ edges, and let G be such a graph with m edges. Furthermore, let u and v be the two odd vertices in G, let $k = \delta(u)$ and let $x_1, \ldots, x_k$ be the neighbours of u. For $j \in \{1, \ldots, k\}$ let $e_j = \{u, x_j\}$ and let T_j be the set of all the trails $u\, e_j\, x_j \ldots\, v$ with the property that v appears only at the end of the trail. Then $T_1 \cup \ldots \cup T_k$ is the set of all the trails we are considering and we have to show that $|T_1| + \ldots + |T_k|$ is odd. Since k is odd, it suffices to show that every $|T_j|$ is odd.

Take any $j \in \{1, \ldots, k\}$ and let $G_j = G - e_j$. The degree of u in G_j is even, so x_j and v are the only odd vertices in G_j. This is why they have to belong to the same connected component of G_j. The number of edges in this connected component is strictly less than m, so by the induction hypothesis the number of trails that start at x_j, end in v and contain v only once is odd. It is easily seen that the number of such trails equals $|T_j|$, and hence $|T_j|$ is also odd. This completes the proof of the claim.

Let us now go back to the proof of the theorem.

($\Leftarrow$) Let G be a connected graph that is not Eulerian. Then G has an odd vertex v. For an edge e incident to v let $c(e)$ denote the number of cycles in G that contain e. Since each such cycle contains two edges that are adjacent to v, the sum $\sum_{v \in e} c(e)$ is even (= twice the number of cycles that pass through v). But $\delta(v)$ is odd, so this sum consists of an odd number of summands. Therefore, one of the summands has to be even, and thus there exists an edge e adjacent to v such that $c(e)$ is even.

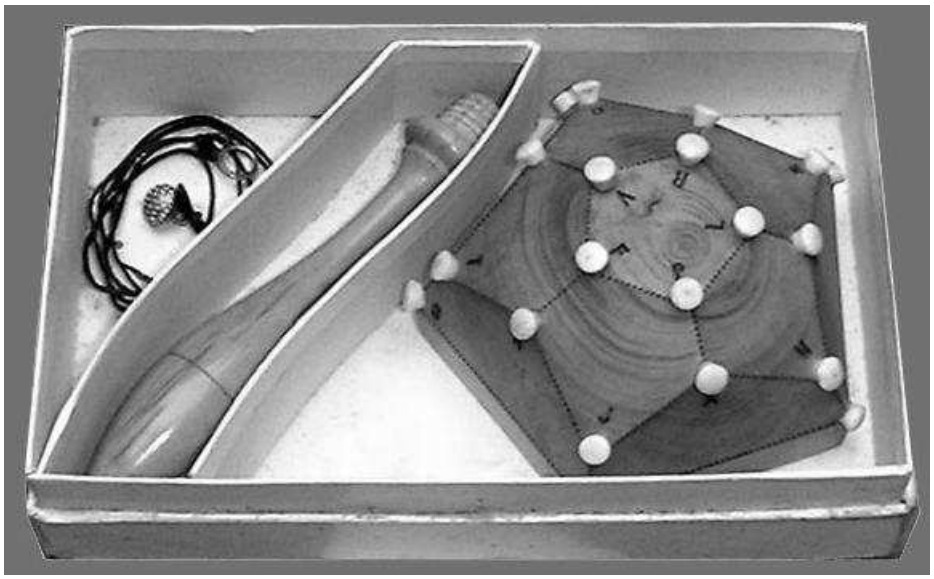
($\Rightarrow$) Let G be an Eulerian graph and let $e = \{u, v\} \in E(G)$ be arbitrary. According to Exercise 3.13, e is not a cut-edge, so $G - e$ is connected. Hence, $G - e$ is not Eulerian, but has an Eularian trail. Let this trail start at u and end in v. The Claim now yields that there is an odd number of trails that start at u, end in v and contain v only once. If S is one such trail which is not a path, then S contains some vertex more than once (for otherwise S would be a path). Let w_i be the first vertex in S that appears more than once in S and let $w_i e_{i+1} w_{i+1} \ldots e_j w_j = w_i$ be the shortest cycle in S that contains w_i. "Mirroring" the cycle within S produces a new trail S' having the same properties as S:

$$\begin{array}{rccccccccccc} S: & u\, e_1\, w_1 \ldots w_i & e_{i+1} & w_{i+1} & \ldots & e_j & w_j \ldots w_{s-1} & e_s & v \\ & & \| & & & & \| & & \\ S': & u\, e_1\, w_1 \ldots w_j & e_j & \ldots & w_{i+1} & e_{i+1} & w_i \ldots w_{s-1} & e_s & v \end{array}$$

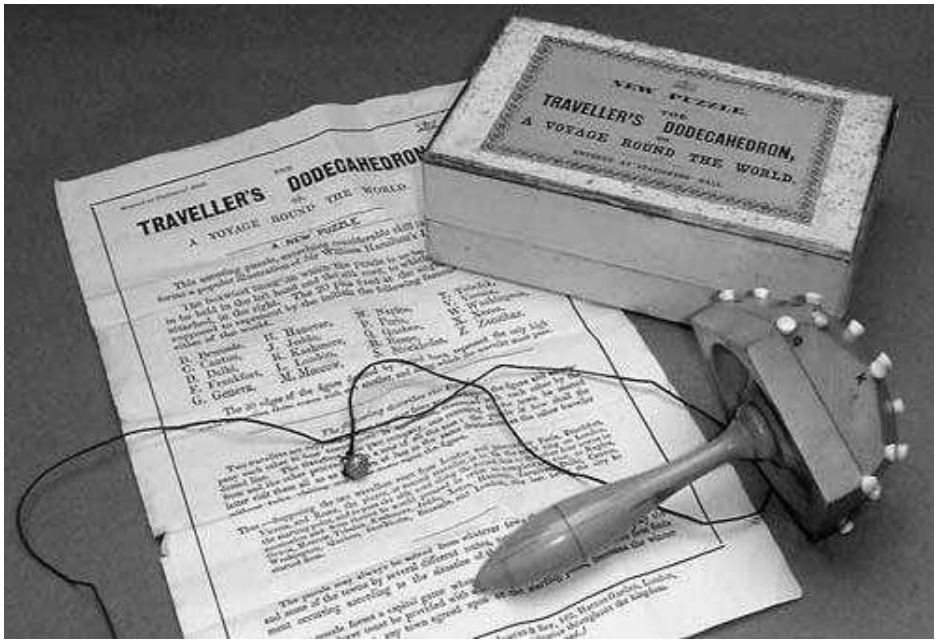
Therefore, trails that start at u, end in v, contain v only once and are not paths appear in pairs. Hence, the number of such trails which are not paths is even. But, we know that there is an odd number of trails with these properties, whence follows that the number of paths connecting u and v in $G - e$ is odd. Each of the paths together with e builds a cycle in G that contains e. Therefore, e belongs to an odd number of cycles. □

3.2 Hamiltonian graphs

Sir William Rowan Hamilton, who was Astronomer Royal of Ireland, invented in 1857 a puzzle called *The Travellers Dodecahedron or A Voyage Around the World*. It is not a true dodecahedron but is a "schematic" of a dodecahedron on a wooden "mushroom".



The 30 edges represent the only roads that one is allowed to pass along as one visits the 20 vertices that represent cities. Two travellers were supposed to set off visiting the cities: the first was supposed to pose a problem and start the tour by visiting four cities that belong to the same face of the dodecahedron. The player posing the problem then returns home and the other continues to travel around the world trying to visit all the remaining cities only once, and eventaully return home. The silk cord that accompanied the puzzle was used to mark the voyage and thus prevent the voyager from visiting a city more than once.



Until recently, only information we had on *The Travellers Dodecahedron* was its description in a chapter on Hamilton's Game in volume 2 of Édouard Lucas' *Récréations Mathématiques* and another mention in the 3rd edition of Ahrens' German work on Recreational Mathematics. But then an example was recovered, complete and in almost new condition.

In graph-theoretic terms the puzzle boils down to finding a spanning cycle of the incidence graph of a dodecahedron. The graph shown in Fig. 3.2 is a plane projection of a dodecahedron and we outlined a spanning cycle in this graph.

Definition 3.8 A *Hamiltonian path* in a graph is a path that contains all vertices of the graph. A *Hamiltonian cycle* in a graph is a cycle that contains all vertices of the graph. A graph is called *Hamiltonian* if it has a Hamiltonian cycle.

In comparison with Eulerian graphs, Hamiltonian graphs are much more hard to grasp. There is no "useful" characterisation of Hamiltonian graphs and we shall see in the next section that there is a justification for this: deciding whether a graph is Hamiltonian is one of the most complicated computational problems. We will actually show that this decision problem is NP-complete (for the moment, think of this as "extremely hard").

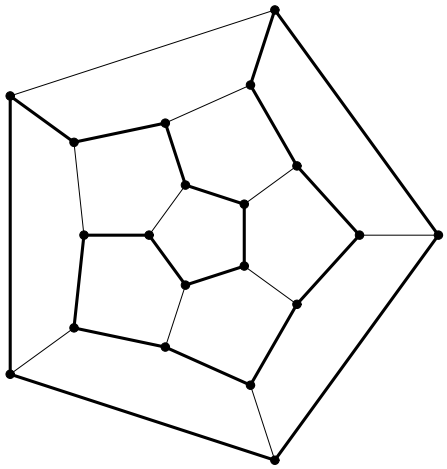


Figure 3.2: A solution to The Travellers Dodecahedron is a spaning cycle of the incidence graph of the dodecahedron

Theorem 3.9 *Let G be a Hamiltonian graph and $\varnothing \neq S \subset V(G)$ a nonempty set of vertices of G. Then $\omega(G-S) \leqslant |S|$.*

Proof. Let C be a Hamiltonian cycle of G. Then $\omega(C-S) \geqslant \omega(G-S)$ since $G-S$ has more edges than $C-S$, and they might connect some of the connected components of $C-S$ together. On the other hand, it is easy to see that $\omega(C-S) \leqslant |S|$. Therefore, $\omega(G-S) \leqslant |S|$. □

Theorem 3.9 is useful when it comes to showing that a graph is *not* Hamiltonian.

Corollary 3.10 *Hamiltonian graphs have no cut-vertices and no cut-edges.*

Proof. If v is a cut-vertex of a graph G then $\omega(G-v) \geqslant 2 > |\{v\}|$. Theorem 3.9 now implies that G is not Hamiltonian. We leave the cut-edges as Homework 3.5. □

We have already mentioned that there is no "useful" characterisation of Hamiltonian graphs. However, it is generally accepted that the best characterization of Hamiltonian graphs was given in 1972 by Bondy and Chvátal who generalized earlier results by G. A. Dirac and O. Ore. The idea behind their result is that a graph is Hamiltonian if enough edges exist.

If u,v are nonadjacent vertices in G and $e = \{u,v\}$, then by $G+e$ we denote the graph obtained by adding the edge e to G.

The closure of a graph G is a graph on the same set of vertices constructed as follows. Define a sequence of graphs $G_0, G_1, \ldots$, by $G_0 = G$ and

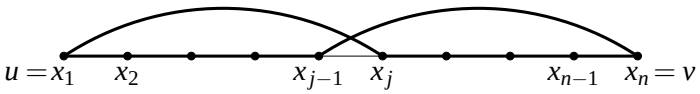
$$G_{i+1} = \begin{cases} G_i + e, & \text{where } e \notin E(G_i) \text{ joins two nonadjacent vertices} \\ & u, v \in V(G_i) \text{ such that } \delta_{G_i}(u) + \delta_{G_i}(v) \geqslant n(G_i), \\ G_i, & \text{if no such pair of vertices exists.} \end{cases}$$

Since we leave the set of vertices fixed and add new edges whenever possible, there exists a k such that $G_k = G_{k+j}$ for all $j \geqslant 1$. Then the graph G_k is called the *closure* of G and denoted by $\mathrm{cl}(G)$.

Theorem 3.11 (Bondy, Chvátal 1972) *A graph G is Hamiltonian if and only if $\mathrm{cl}(G)$ is Hamiltonian.*

Proof. If G is Hamiltonian, then so is $\mathrm{cl}(G)$ since $E(G) \subseteq E(\mathrm{cl}(G))$. For the converse, suppose that G is not Hamiltonian but that $\mathrm{cl}(G)$ is Hamiltonian. Then there exists a graph G_i in the sequence $G = G_0, G_1, \ldots, G_k = \mathrm{cl}(G)$ defining $\mathrm{cl}(G)$ such that G_i is not Hamiltonian and G_{i+1} is Hamiltonian. Let $G_{i+1} = G_i + e$ where $e = \{u, v\}$. Then by the construction, u and v are not adjacent and $\delta_{G_i}(u) + \delta_{G_i}(v) \geqslant n$.

Since $G_i + e$ is Hamiltonian and G_i is not, it follows that each Hamiltonian cycle in $G_i + e$ passes through e. Take any Hamiltonian cycle C in $G_i + e$. Then $e \in E(C)$ and hence $C - e$ is a Hamiltonian path $u = x_1\, x_2\, \ldots\, x_{n-1}\, x_n = v$ in G_i. Now it is easy to see that if u is adjacent to x_j for some $j > 1$ then v is *not* adjacent to x_{j-1} for otherwise we would have a Hamiltonian cycle in G_i:



Therefore, if $\delta_{G_i}(u) = k$ then $\delta_{G_i}(v) \leqslant n - (1 + k)$ since v is not adjacent to itself, nor is it adjacent to predecessors of the k neighbours of u. Hence $\delta_{G_i}(u) + \delta_{G_i}(v) \leqslant n - 1$. Contradiction. □

Corollary 3.12 *Let G be a graph with n vertices.*

(a) *If $\delta(u)+\delta(v) \geqslant n$ whenever u and v are distinct, nonadjacent vertices of G then G is Hamiltonian. (O. Ore 1960)*

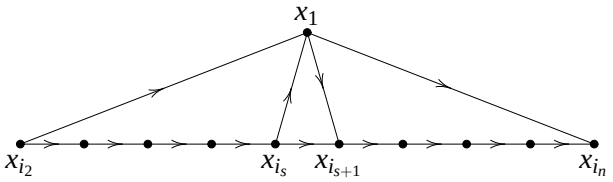
(b) *If $\delta(u) \geqslant \frac{n}{2}$ for all $u \in V(G)$ then G is Hamiltonian. (G. A. Dirac 1952)*

All these statements have their analogues for digraphs. We shall, however, treat only tournaments to show how very special digraphs they are.

Definition 3.13 A *Hamiltonian path* in a digraph is an oriented path that contains all vertices of the digraph. A *Hamiltonian cycle* in a digraph is an oriented cycle that contains all vertices of the digraph. A digraph is called *Hamiltonian* if it has a Hamiltonian cycle.

Theorem 3.14 (Rédei) *Every tournament has a Hamiltonian path.*

Proof. The proof is by induction on the number of vertices in the tournament. The statement is easily seen to be true in case of tournaments with 2 and 3 vertices. Assume now that every tournament with less than n vertices has a Hamiltonian path, and let T be a tournament on n vertices, $V(T)=\{x_1,\ldots,x_n\}$. By the induction hypothesis $T'=T-x_1$ has a Hamiltonian path $x_{i_2}\,x_{i_3}\,\ldots\,x_{i_n}$. If $x_1 \to x_{i_2}$ or $x_{i_n} \to x_1$, the Hamiltonian path of T' easily extends to a Hamiltonian path of T. If, however, $x_1 \not\to x_{i_2}$ and $x_{i_n} \not\to x_1$ then $x_{i_2} \to x_1$ and $x_1 \to x_{i_n}$. It is easy to see that there exists an s such that $x_{i_s} \to x_1 \to x_{i_{s+1}}$:



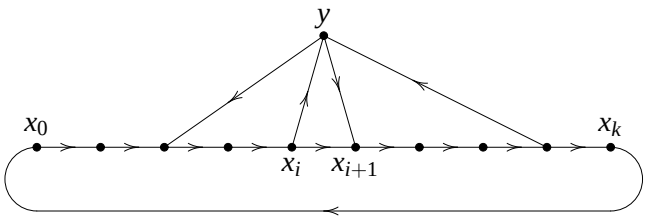
so $x_{i_2}\,\ldots\,x_{i_s}\,x_1\,x_{i_{s+1}}\,\ldots\,x_{i_n}$ is a Hamiltonian path for T. □

Theorem 3.15 *A tournament is Hamiltonian if and only if it is strongly connected.*

Proof. ($\Rightarrow$) If a tournament is Hamiltonian, then walking along the Hamiltonian cycle we can get from every vertex of the tournament to every other vertex. Hence, the tournament is strongly connected.

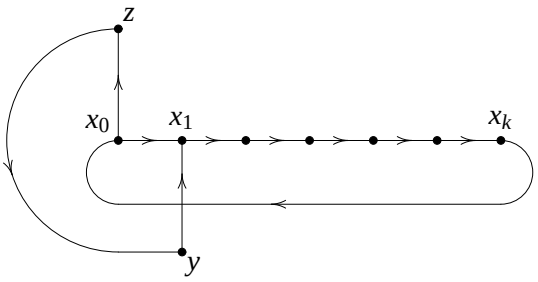
($\Leftarrow$) Let T be a strongly connected tournament. Then T is not transitive and hence contains an oriented cycle. Let $C=x_0 \to x_1 \to \ldots \to x_k \to x_0$ be the longest

oriented cycle in T and let us show that $V(C) = V(T)$. Suppose to the contrary that $V(C) \subset V(T)$. Then $\{V(C), B\}$ is a partition of $V(T)$, where $B = V(T) \setminus V(C)$. If there exists a $y \in B$ such that $E(V(C), \{y\}) \neq \varnothing$ and $E(\{y\}, V(C)) \neq \varnothing$ then there exists an index i such that $x_i \to y \to x_{i+1}$:



and $x_0 \to \ldots \to x_i \to y \to x_{i+1} \to \ldots \to x_k \to x_0$ is an oriented cycle in T which is longer than C. Contradiction.

Therefore, for each $y \in B$ either $E(V(C), \{y\}) = \varnothing$ or $E(\{y\}, V(C)) = \varnothing$. Let $Y = \{y \in B : E(V(C), \{y\}) = \varnothing\}$ and $Z = \{z \in B : E(\{z\}, V(C)) = \varnothing\}$. Since T is strongly connected it follows that $Y \neq \varnothing$, $Z \neq \varnothing$ and $E(Z, Y) \neq 0$. Take $z \in Z$ and $y \in Y$ such that $z \to y$. From $E(V(C), \{y\}) = \varnothing$ it follows that $y \to x_i$ for all i.



Similarly, $x_i \to z$ for all i, so $x_0 \to z \to y \to x_1 \to \ldots \to x_k \to x_0$ is an oriented cycle in T and it is longer than C. Contradiction. Therefore, $V(C) = V(T)$, so T is a Hamiltonian tournament. □

A careful analysis of the previous proof reveals that we can actually prove much more.

Theorem 3.16 (Camion 1959) *Let T be a Hamiltonian tournament with n vertices. For every vertex $v \in V(T)$ and every $k \in \{3, \ldots, n\}$ there exists an oriented cycle of length k that contains v.*

3.3 Complexity issues

In this section we consider the computational complexity of deciding whether a graph has a Hamiltonian cycle. We show that this decision problem not only falls into the NP complexity class, but that it is an NP-complete problem, i.e. a paradigm of an NP-hard problem.

The notion of an algorithm (= "effective procedure") was recognised as one of the essential notions in mathematics as early as 1928 when D. Hilbert and W. Ackermann published their influential booklet "Grundzüge der theoretischen Logik" in which they posed a problem of finding an algorithm (whatever that might mean) which decides whether a first-order sentence is a consequence of the axioms of arithmetic. At that time there was no formal notion of an algorithm, so the problem was actually twofold: on the "philosophical" level it was required to introduce the precise definition of an algorithm, while on the mathematical level the definition should have been used in solving the particular problem of mathematical logic. The problem (both on the philosophical and the mathematical level) was independently solved in 1936 by A. Church and A. Turing. Although Church's solution was published a few months ahead of Turing's, the approach taken by A. Turing is more intuitive, and constitues a basis of what is today known as Computability Theory.

We shall not present a formal definition of a Turing machine. For our purposes it suffices to say that a *Turing machine* is a mathematical model of a computer program written for a modern computer with infinite memory. Since computers actually operate on finite 01-words we shall take $\Sigma = \{0,1\}$ as the alphabet in which to carry out our considerations. Let Σ^* denote the set of all finite 01-words, together with the empty word ε. By $|w|$ we denote the length of $w \in \Sigma^*$. A *language* is any set $\mathscr{L} \subseteq \Sigma^*$ of 01-words. In particular, for every graph G there is a 01-word $\langle G \rangle$ representing the graph, so we also have the language $\mathscr{G} = \{\langle G \rangle : G \text{ is a graph}\}$.

A computer program A can take any 01-word w as its input, but may fail to produce an output. Hence, each computer program A corresponds to a function $\widehat{A} : \Sigma^* \to \Sigma^* \cup \{\infty\}$ such that

$$\widehat{A}(w) = \begin{cases} u, & A \text{ takes } w \text{ as its input and after a finite number of computation} \\ & \text{steps stops and prints } u \text{ as a result;} \\ \infty, & A \text{ never stops on input } w. \end{cases}$$

For a computer program A and a word $w \in \Sigma^*$ let

$$t_A(w) = \begin{cases} n, & A \text{ takes } w \text{ as its input and stops after } n \text{ computation steps;} \\ \infty, & A \text{ never stops on input } w. \end{cases}$$

A computer program A *runs in polynomial time* if there exists a positive integer k such that $t_A(w) = O(|w|^k)$ whenever $\widehat{A}(w) \neq \infty$.

The complexity class P. A language $\mathscr{L} \subseteq \Sigma^*$ is *decidable* if there exists a computer program A such that $\widehat{A} : \Sigma^* \to \{0,1\}$ and

$$\mathscr{L} = \{w \in \Sigma^* : \widehat{A}(w) = 1\}.$$

(Note that the computer program which decides a language stops on all inputs and outputs 0 or 1.) The language $\mathscr{L} \subseteq \Sigma^*$ is *decidable in polynomial time* if there exists a computer program A wich runs in polynomial time such that $\widehat{A} : \Sigma^* \to \{0,1\}$ and $\mathscr{L} = \{w \in \Sigma^* : \widehat{A}(w) = 1\}$.

Definition 3.17 The complexity class P consists of all languages over $\Sigma = \{0,1\}$ that are decidable in polynomial time:

$$\mathsf{P} = \{\mathscr{L} \subseteq \Sigma^* : \mathscr{L} \text{ is decidable in polynomial time}\}.$$

Equivalently, the complexity class P consists of all problems that can be solved in polynomial time. Indeed, given an problem Q it suffices to encode each instance I of the problem by a 01-word $\langle I \rangle$ and consider the language $\mathscr{L}_Q = \{\langle I \rangle : I \text{ is an instance of } Q\}$. Then each instance I of the problem can be solved in polynomial time (where the degree of the polynomial does not depend on the instance) if and only if $\mathscr{L}_Q$ is decidable in polynomial time. For example, the problem of deciding in polynomial time whether a graph is connected corresponds to polynomial decidability of the language $\mathscr{L}_{conn} = \{\langle G \rangle : G \text{ is a connected graph}\}$. For some other problems the transformation $\boxed{\text{problem}} \to \boxed{\text{language}}$ may not be so obvious.

The complexity class NP. Instead of requiring a computer program to solve a problem, we might only wish to pull a solution out of a sleeve and verify that then solution is indeed a solution to a problem. A *verification algorithm* is a computer program A with two inputs such that $\widehat{A} : \Sigma^* \times \Sigma^* \to \{0,1\}$. If there exists a positive integer k such that $t_A(p,s) = O((|p| + |s|)^k)$ for all $p, s \in \Sigma^*$ we say that A *is a polynomial verification algorithm*. A language $\mathscr{L}$ is *verified* by a verification algorithm A if

$$\mathscr{L} = \{p \in \Sigma^* : \exists s \in \Sigma^* \ (\widehat{A}(p,s) = 1)\}.$$

A *language* $\mathscr{L} \subseteq \Sigma^*$ *is verifiable in polynomial time* if there exists a positive integer c and a polynomial verification algorithm A such that

$$\mathscr{L} = \{p \in \Sigma^* : \exists s \in \Sigma^* \ (|s| \leqslant |p|^c \text{ and } \widehat{A}(p,s) = 1)\}.$$

Definition 3.18 The complexity class NP consists of all languages over $\Sigma = \{0,1\}$ that are verifiable in polynomial time:

$$\mathsf{NP} = \{\mathscr{L} \subseteq \Sigma^* : \mathscr{L} \text{ is verifiable in polynomial time}\}.$$

Equivalently, the complexity class NP consists of problems for which it is easy to check whether what we claim to be a solution is indeed a solution. For example, $\mathscr{L}_{Ham} = \{\langle G\rangle : G \text{ is a Hamiltonian graph}\}$ is in NP since given a graph G and a sequence of vertices $x_1,\ldots,x_n$ it is easy to check whether $x_1,\ldots,x_n$ is a Hamiltonian cycle of G.

Theorem 3.19 $\mathsf{P} \subseteq \mathsf{NP}$.

Proof. Take any $\mathscr{L} \in \mathsf{P}$. Then $\mathscr{L} = \{w \in \Sigma^* : \widehat{A}(w) = 1\}$ for some computer program A that decides $\mathscr{L}$ in polynomial time. Now take a verification algorithm $B : \Sigma^* \times \Sigma^* \to \{0,1\}$ so that $\widehat{B}(p,s) = \widehat{A}(p)$. Then B clearly verifies $\mathscr{L}$ in polynomial time, so $\mathscr{L} \in \mathsf{NP}$. □

The exact relationship between P and NP is still unknown. It is strongly believed that $\mathsf{P} \neq \mathsf{NP}$, but we still haven't got a proof. The problem is actually so important that the Clay Mathematics Institute is offering a USD 1,000,000 prize for the correct solution.[1] Apart from the prize, the importance of the problem is also reflected by the fact that the security of RSA, the most widely used crypto-system, depends on $\mathsf{P} \neq \mathsf{NP}$. If it turns out that $\mathsf{P} = \mathsf{NP}$ the security of all transactions based on RSA, PGP and the such will be broken and many aspects of our everyday life would have to change.

Polynomial reducibility and NP-completeness. We say that a language $\mathscr{L}_1 \subseteq \Sigma^*$ is *polynomially reducible* to a language $\mathscr{L}_2 \subseteq \Sigma^*$, and write $\mathscr{L}_1 \preccurlyeq_p \mathscr{L}_2$, if there exists a computer program A which runs in polynomial time such that $\widehat{A} : \Sigma^* \to \Sigma^*$ and

$$w \in \mathscr{L}_1 \text{ if and only if } \widehat{A}(w) \in \mathscr{L}_2.$$

Intuitively, regarding polynomial-time as "easy", this means: if there is a polynomial reduction from $\mathscr{L}_1$ to $\mathscr{L}_2$, then $\mathscr{L}_1$ cannot be harder than $\mathscr{L}_2$.

Theorem 3.20 *If* $\mathscr{L} \in \mathsf{P}$ *and* $\mathscr{L}' \preccurlyeq_p \mathscr{L}$ *then* $\mathscr{L}' \in \mathsf{P}$.

[1] http://www.claymath.org/millennium/P_vs_NP/

Proof. If A is a computer program that decides $\mathscr{L}$ in polynomial time, and if B is a computer program that reduces $\mathscr{L}'$ to $\mathscr{L}$ in polynomial time, then $B \circ A$ is a computer program that decides $\mathscr{L}'$ in polynomial time, so $\mathscr{L}' \in \mathsf{P}$. □

Definition 3.21 A language $\mathscr{L} \subseteq \Sigma^*$ is NP-*hard* if $\mathscr{L}' \preccurlyeq_p \mathscr{L}$ for every $\mathscr{L}' \in \mathsf{NP}$. A language $\mathscr{L} \subseteq \Sigma^*$ is NP-*complete* if it is NP-hard and belongs to NP.

An NP-complete problem is a paradigm of an NP-problem. Moreover, if one of them happens to be in P then $\mathsf{P} = \mathsf{NP}$:

Theorem 3.22 *Let $\mathscr{L}$ be an* NP*-complete language. If $\mathscr{L} \in \mathsf{P}$ then* $\mathsf{P} = \mathsf{NP}$.

Proof. Suppose that $\mathscr{L}$ is an NP-complete language such that $\mathscr{L} \in \mathsf{P}$. Take any $\mathscr{L}' \in \mathsf{NP}$. Since $\mathscr{L}$ is NP-hard, it follows that $\mathscr{L}' \preccurlyeq_p \mathscr{L}$ and thus $\mathscr{L}' \in \mathsf{P}$ by Theorem 3.20. This shows that $\mathsf{NP} \subseteq \mathsf{P}$. □

The first hands-on NP-complete problem was discovered in 1971 by S. Cook. A *Boolean formula* is a formula built up from Boolean variables $x_1, \ldots, x_n$ (each of which can take the values *true* or *false*) and Boolean connectives $\neg$, $\wedge$ and $\vee$. A Boolean formula $F(x_1,\ldots,x_n)$ is said to be in a *conjunctive form* (CF for short) if it has the form

$$F(x_1,\ldots,x_n) = C_1(x_1,\ldots,x_n) \wedge C_2(x_1,\ldots,x_n) \wedge \ldots \wedge C_k(x_1,\ldots,x_n)$$

where each clause $C_i(x_1,\ldots,x_n)$ is a disjuction of literals

$$C_i(x_1,\ldots,x_n) = (l_{i1} \vee l_{i2} \vee \ldots \vee l_{im_i})$$

and each literal l_{ij} is a variable x_{ij} or a negated variable $\neg x_{ij}$. It is a well known fact from Boolean logic that every Boolean formula is equivalent to a CF Boolean formula.

A Boolean formula $F(x_1,\ldots,x_n)$ is *satisfiable* if there exists an assignment $\tau : \{x_1,\ldots,x_n\} \to \{true, false\}$ of truth values to variables such that $\tau(F) = true$, that is, F evaluates to *true* under the assignment τ. Let us fix a systematic way of encoding CF Boolean formulas by 01-words and let $\langle F \rangle$ denote an encoding of F. Let us denote the language that corresponds to satisfiable Boolean formulas by *SAT*:

$$SAT = \{\langle F \rangle : F \text{ is a satisfiable CF Boolean formula}\}.$$

Theorem 3.23 (Cook 1971) *SAT is* NP*-complete.*

Now that we have an explicite NP-complete problem, it gives us a strategy to show that other problems are also NP-complete: if an NP-complete problem is polynomially reducible to some other problem, this new problem also has to be NP-complete.

Theorem 3.24 *If $\mathscr{L}$ is an* NP*-complete language and if $\mathscr{L}' \in$ NP has the property that $\mathscr{L} \preccurlyeq_p \mathscr{L}'$ then $\mathscr{L}'$ is also* NP*-complete.*

Proof. This is an immediate consequence of the fact that $\preccurlyeq_p$ is transitive. □

Therefore, in order to show that finding a Hamiltonian cycle in a graph is an NP-complete problem, it suffices to show that *SAT* is polynomially reducible to it. In this particular case, working with digraphs turns out to be easier than working with graphs, so we introduce the two languages:

- $HAMG = \{\langle G\rangle : G \text{ is a Hamiltonian graph}\}$, which is a 01-language that encodes Hamiltonian graphs, and
- $HAMD = \{\langle D\rangle : D \text{ is a Hamiltonian digraph}\}$, which is a 01-language that encodes Hamiltonian digraphs,

and carry out the proof in two steps:

- we first show that $HAMG \preccurlyeq_p HAMD$ and $HAMD \preccurlyeq_p HAMG$; and then
- we show that $SAT \preccurlyeq_p HAMD$.

Lemma 3.25 *$HAMG \preccurlyeq_p HAMD$ and $HAMD \preccurlyeq_p HAMG$.*

Proof. For every graph $G = (V,E)$ let $D_G = (V,E')$ denote the digraph with the same set of vertices whose set of edges is

$$E' = \{(u,v) \in V^2 : \{u,v\} \in E\}.$$

Clearly, there exists a polynomial algorithm that converts $\langle G\rangle$ to $\langle D_G\rangle$ and it is easy to see that G is a Hamiltonian graph if and only if D_G is a Hamiltonian digraph (Homework 3.11). Therefore, $HAMG \preccurlyeq_p HAMD$.

Now, let $D = (V,E)$ be a digraph and let $G_D = (V',E')$ be a graph constructed from D as follows. For each $v \in V$ we add three vertices v^0, v^1, v^2 to V' and two edges $\{v^0, v^1\}$ and $\{v^1, v^2\}$ to E' replacing thus each vertex of D by a path of length 2 in G_D. Moreover, for each edge (u,v) in E we add an edge $\{u^2, v^0\}$ to E'. An illustration of this proces is given in Fig. 3.3. Clearly, $|V'| = 3|V|$ and $|E'| = |E| + 2|V|$, so the reduction is polynomial. It is also easy to see that D is a

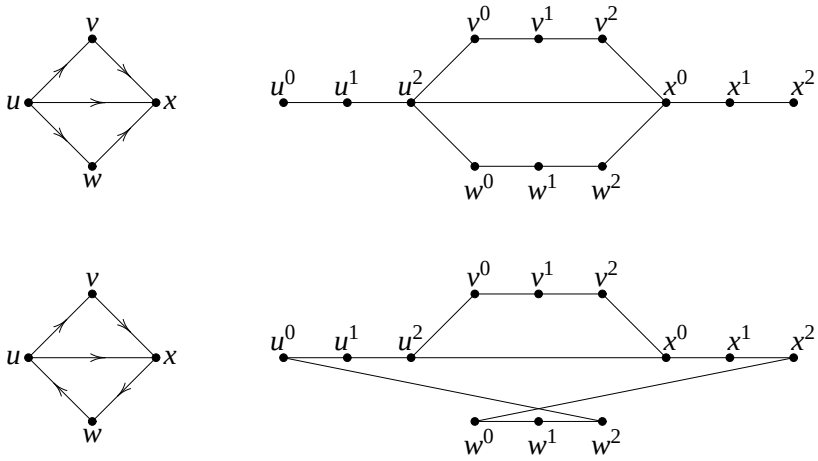


Figure 3.3: Two digraphs and their associated graphs

Hamiltonian digraph if and only if G_D is a Hamiltonian graph (Homework 3.11). Therefore, *HAMD* $\preccurlyeq_p$ *HAMG*. □

Theorem 3.26 *HAMG is* NP*-complete.*

Proof. According to Theorem 3.24 it suffices to show that *SAT* $\preccurlyeq_p$ *HAMG*. We shall actually show that *SAT* $\preccurlyeq_p$ *HAMD* and then use *HAMD* $\preccurlyeq_p$ *HAMG* established in Lemma 3.25. Therefore, for every Boolean formula $F(x_1,\ldots,x_n)$ in CF we have to construct a not too complicated digraph D_F such that F is satisfiable if and only if D_F has an oriented Hamiltonian cycle.

Let $F(x_1,\ldots,x_n)$ be a Boolean formula given in its conjunctive form:

$$F(x_1,\ldots,x_n) = C_1(x_1,\ldots,x_n) \wedge C_2(x_1,\ldots,x_n) \wedge \ldots \wedge C_k(x_1,\ldots,x_n).$$

Recall that each clause $C_i(x_1,\ldots,x_n)$ is a disjuction of literals

$$C_i(x_1,\ldots,x_n) = (l_{i1} \vee l_{i2} \vee \ldots \vee l_{im_i})$$

and each literal l_{ij} is a variable x_{ij} or a negated variable $\neg x_{ij}$. We construct a digraph D_F with $2nk+k$ vertices as follows. For each variable x_i we have $2k$ vertices u_{i1}, v_{i1}, u_{i2}, v_{i2}, …, u_{ik}, v_{ik}, and for each clause C_i we have a vertex c_i. The vertices u_{ij}, v_{ij} are connected by edges as in Fig. 3.4. We choose a direction, say from left to right, and say that that x_i evaluates to *true* if we traverse vertices that correspond to x_i in that direction, while it evaluates to false if we traverse the vertices that correspond to x_i in the oposite direction.

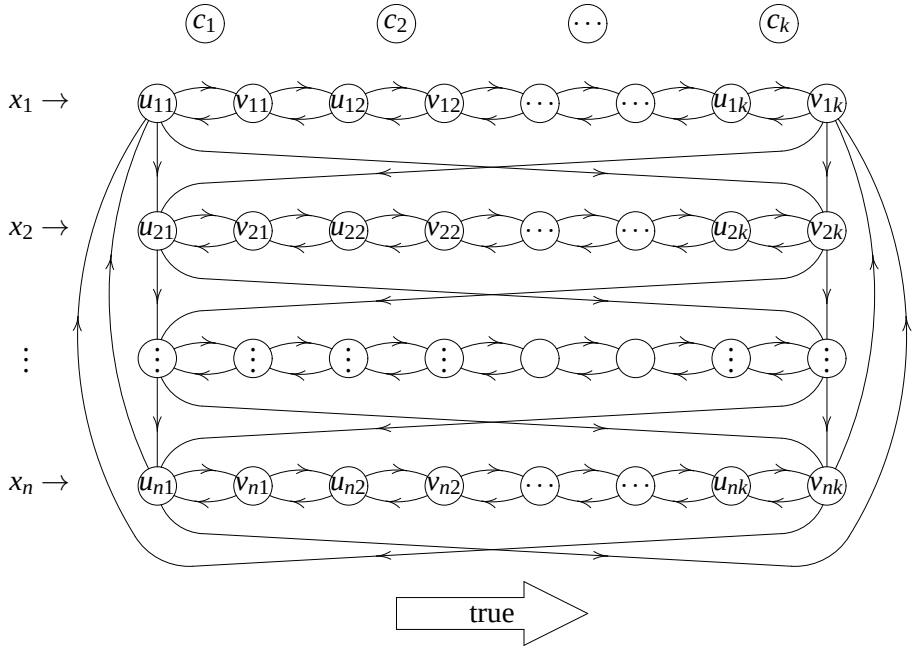


Figure 3.4: The construction of the digraph D_F, Part I: vertices that correspond to variables

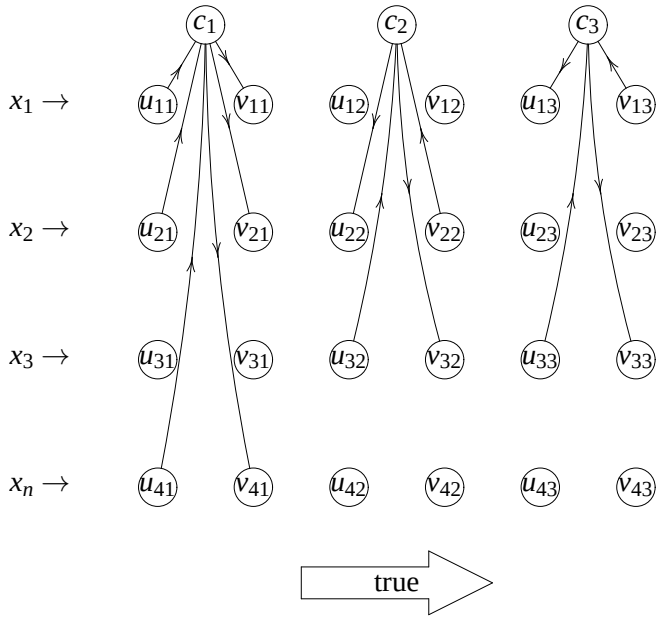


Figure 3.5: The construction of the digraph D_F, Part II: vertices that correspond to clauses

Next, we describe how to connect vertices that correspont to clauses to vertices that correspond to variables. If a variable x_i appears in a clause C_j and it is not negated in C_j, we add the edges $u_{ij} \rightarrow c_j$ and $c_j \rightarrow v_{ij}$. If, however, x_i is negated in C_j we add the edges $v_{ij} \rightarrow c_j$ and $c_j \rightarrow u_{ij}$. So, if a variable x_i is not negated in a clause C_j we add edges that go "in the direction of truth". If x_i is negated in C_j, we add edges that go "in the direction oposite of truth". An example is given in Fig. 3.5 (for clarity, the figure indicates only the edges incident to vertices that represent clauses; edges connecting u_{ij}'s to v_{ij}'s have been omitted). The digraph in Fig. 3.5 corresponds to the boolean formula $F(x_1,x_2,x_3,x_4) = C_1 \wedge C_2 \wedge C_3$ where $C_1 = x_1 \vee x_2 \vee x_4$, $C_2 = \neg x_2 \vee x_3$ and $C_3 = \neg x_1 \vee x_3$. The full graph that represents F is given in Fig. 3.6.

It is easy to see that this construction can be carried out in polynomial time. Let us finally show that F is satisfiable if and only if D_F has an oriented Hamiltonian cycle. Recall that traversing a row of vertices that corresponds to x_i from left to right means $\tau(x_i) = \mathit{true}$ while traversing from right to left means $\tau(x_i) = \mathit{false}$. The idea is that an oriented Hamiltonian cycle through the digraph represents an assignment of truth values to the variables $x_1, \ldots, x_n$.

Assume the formula F is satisfiable by some truth assignment τ. Choose one true literal in each clause, traverse the graph moving across each variableŠs path

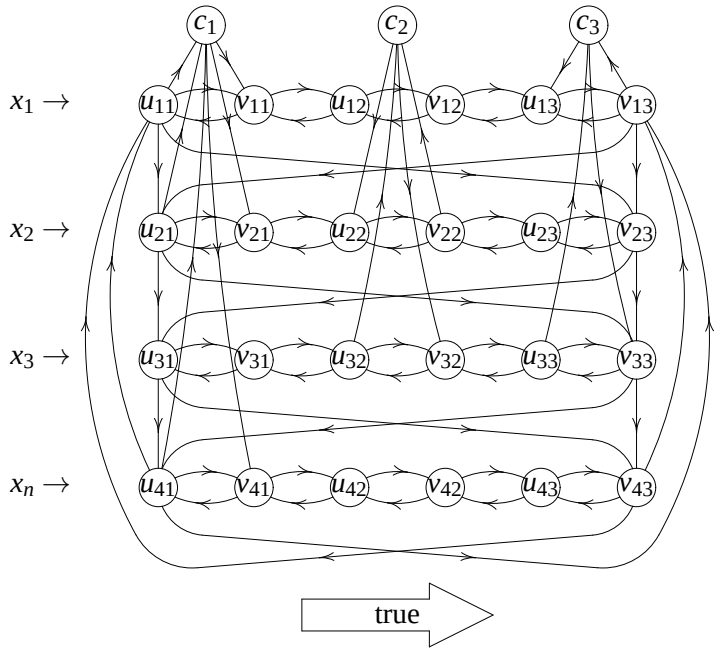


Figure 3.6: The digraph D_F for $F(x_1,x_2,x_3,x_4) = (x_1 \vee x_2 \vee x_4) \wedge (\neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_3)$

in the appropriate direction, and take a diversion to a clause-node for each literal chosen above. This oriented path is a Hamiltonian cycle.

Conversely, suppose there exists an oriented Hamiltonian cycle H in D_F. Then H traverses each variableŠs row either from left to right or from right to left and thus determines an assignment of truth values τ to variables. Each clause-node is visited by a side-trip from a variable row. This variable corresponds to a true literal in the clause. Hence, each clause evaluates to *true* under τ and hence $\tau(F) = \textit{true}$, i.e. F is a satisfiable formula. □

Homework

3.1. Let D be an Eulerian digraph. Prove that each closed Eulerian trail in D can be partitioned into oriented cycles in such a way that every edge of D belongs to exactly one of the cycles. (Hint: use induction on the length of the trail.)

3.2. Prove Theorem 3.5.

3.3. Complete the proof of Theorem 3.6.

3.4. There are five regular polyhedra: tetrahedron, hexahedron, octahedron, dodecahedron and icosahedron (Fig. 3.7). Which of them could have been used instead of the dodecahedron in the Hamilton's Voyage Around the World puzzle?

3.5. Complete the proof of Corollary 3.10.

3.6. Prove Corollary 3.12. (Hint: for (a) show that $\mathrm{cl}(G)$ is a complete graph and use the Bondy-Chvátal Theorem; (b) follows from (a).)

3.7. (Ore 1960) Let G be a graph with n vertices. If $\delta(u) + \delta(v) \geqslant n-1$ whenever u and v are distinct, nonadjacent vertices of G then G has a Hamiltonian path. (Hint: add a new vertex to G and connect it by an edge to every vertex of G; show that the new graph is Hamiltonian using a similar result for Hamiltonian graphs.)

3.8. Show that a transitive tournament has exactly one Hamiltonian path.

3.9. Show that each tournament which is not strongly connected can be turned into a strongly connected tournament by changing the orientation of only one edge.

3.10. Prove Theorem 3.16. (Hint: induction on k using the fact that a Hamiltonian tournament is strongly connected; for $k = 3$ show that $E(O(v), I(v)) \neq$

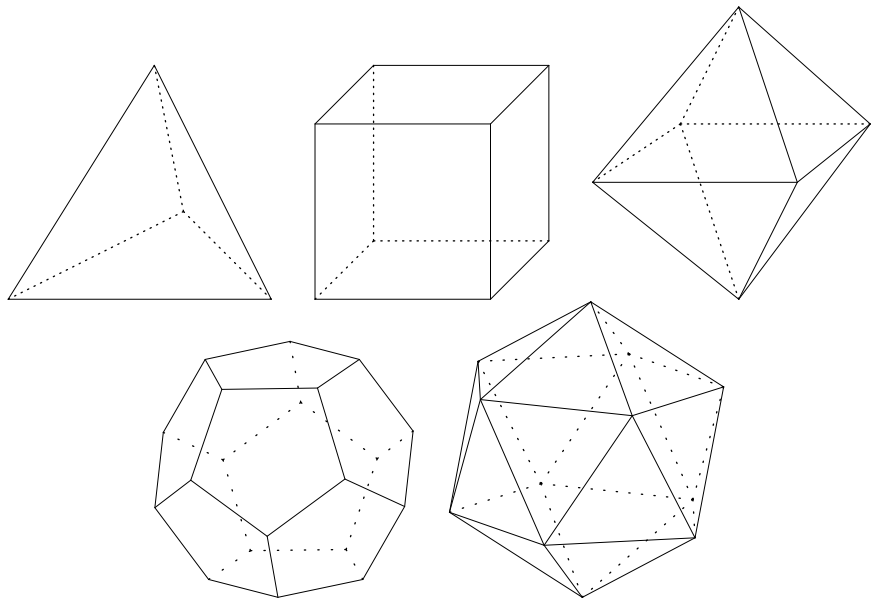


Figure 3.7: The five regular polyhedra

$\varnothing$; for the induction step modify slightly the idea used in the proof of Theorem 3.15.)

3.11. Complete the proof of Lemma 3.25 by showing that

- G is a Hamiltonian graph if and only if D_G is a Hamiltonian digraph; and
- D is a Hamiltonian digraph if and only if G_D is a Hamiltonian graph.

Exercises

3.12. (a) For each $n \geqslant 2$ give an example of a graph with n vertices which is neither Eulerian nor Hamiltonian.

(b) For each $n \geqslant 3$ give an example of a graph with n vertices which is both Eulerian and Hamiltonian.

(c) For each $n \geqslant 4$ give an example of a Hamiltonian graph with n vertices which is not Eulerian.

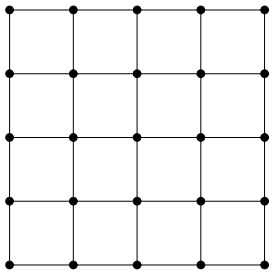


Figure 3.8: Exercise 3.19

(d) For each $n \geqslant 5$ give an example of an Eulerian graph with n vertices which is not Hamiltonian.

3.13. Prove that an Eulerian graph with no isolated vertices has no cut-edges.

3.14. For a digraph D and a set of edges $F \subseteq E(D)$ let W be the set of all vertices of D incident to an edge in F and let $D[F] = (W, F)$ denote the *subdigraph of D induced by F*.

Let D be a weakly connected digraph. Prove that D is Eulerian if and only if there exists a partition $\{F_1, \ldots, F_k\}$ of $E(D)$ such that each $D[F_i]$ is an oriented cycle.

3.15. Let A be a finite set with at least three elements. On $V = \mathscr{P}(A) \setminus \{\varnothing, A\}$ as a set of vertices we define a graph G as follows: two proper subsets X and Y of A are adjacent if and only if $X \subset Y$ or $Y \subset X$ (i.e., if and only if one of them is a proper subset of the other one). Show that G is an Eulerian graph.

3.16. Let G be an Eulerian graph with no isolated vertices and with $n(G)$ odd. If $\Delta(G) \leqslant \lfloor \frac{n}{2} \rfloor$ show that $\overline{G}$ is an Eulerian graph.

3.17. Let G be a connected Eulerian graph with no isolated vertices and with $n(G)$ odd. If $d(G) \geqslant 3$ show that $\overline{G}$ is an Eulerian graph.

3.18. Let G be a connected graph with $2k$ odd vertices. Show that $E(G)$ can be partitioned into k edge-disjoint trails.

3.19. Is it possible to partition the edge-set of the graph in Fig. 3.8 into five edge-disjoint paths of legth 8?

3.20. Which of the graphs in Fig. 3.9 are Hamiltonian?

3.21. (a) Let G be a bipartite Hamiltonian graph and let $\{X, Y\}$ be a partition

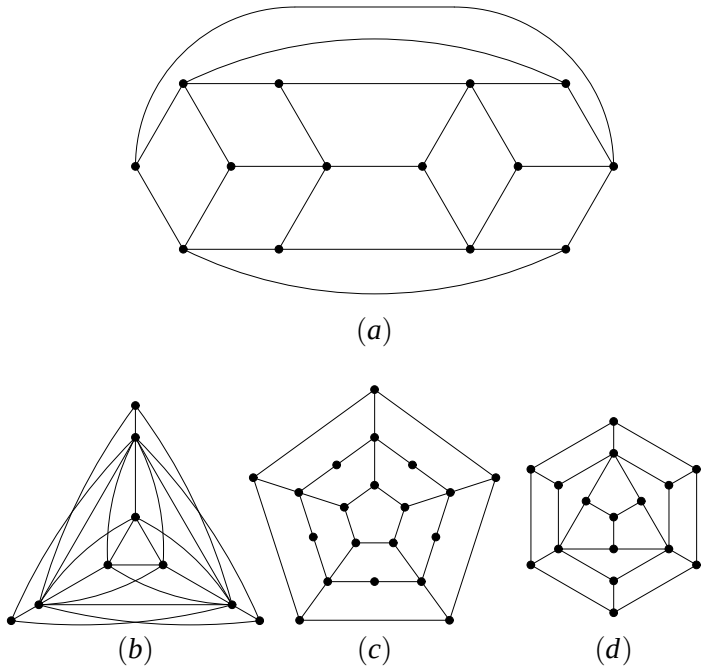


Figure 3.9: Exercise 3.20

of the set of its vertices that demonstrates that G is bipartite. Show that $|X| = |Y|$.

(b) Is the graph in Fig. 3.10 Hamiltonian?

3.22. A vertex cover of a graph G is a set of vertices $W \subseteq V(G)$ such that every edge in G is incident to a vertex from W. Show that if G has a vertex cover W such that $|W| < \frac{1}{2}n(G)$ then G is not Hamiltonian.

3.23. Let G be a graph with n vertices and m edges such that $m \geqslant \binom{n-1}{2} + 2$. Show that G is a Hamiltonian graph.

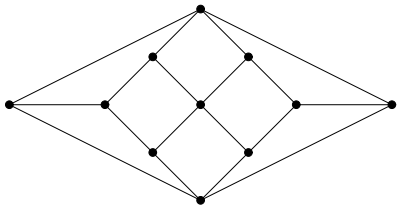


Figure 3.10: Exercise 3.21

3.24. Show that the complement of a regular disconnected graph is a Hamiltonian graph.

3.25. Show that a hypercube of dimension $k \geqslant 2$ is a Hamiltonian graph.

3.26. Show that every strongly connected tournament with $n \geqslant 4$ vertices contains a vertex v such that after changing the orientation of all the edges incident to v we again obtain a strongly connected tournament.

3.27. Show that a strongly connected tournament with $n \geqslant 3$ vertices has at least $n-2$ oriented triangles. (An oriented triange is an oriented cycle of length 3.)

†**3.28.** Let $s_1 \leqslant s_2 \leqslant \ldots \leqslant s_n$ be the scores in a tournament T with n vertices. If $s_n - s_1 < \frac{n}{2}$, show that T is a Hamiltonian tournament. (Hint: show that $s_j - s_i < \frac{n}{2}$ whenever $i < j$ and conclude that T is strongly connected.)

Chapter 4

Introduction to Clones

Boolean logic (or propositional logic as we prefer to call it) is named after George Boole, a professor at University College Cork, who first thought about an algebraic system of logic in the chapter "Of Hypotheticals" of his 1847 book "The Mathematical Analysis of Logic".

The ideas of George Boole (that can be traced back to Leibniz, actually) have reached their final form in the formalisation of mathematical logic at the beginning of the 20th century, which, among other things, lead to the clear distinction between the *syntax* and the *semantics* of logical systems. In case of Boolean logic, the syntactic part consists of propositional formulas, while the semantics is provided by Boolean functions, e.g.

$$p \vee (q \wedge \neg r) \qquad \text{vs.}$$

p	q	r	$f(p,q,r)$
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

It is an easy obsetvation that every Boolean function $f(x_1,\ldots,x_n)$ in n unknowns which is not identically equal to 0 can be represented by a formula of the propositional calculus as follows:

$$f(x_1,\ldots,x_n) \quad = \bigvee_{\substack{\varepsilon_1,\ldots,\varepsilon_n:\\ f(\varepsilon_1,\ldots,\varepsilon_n)=1}} x_1^{\varepsilon_1} \wedge \ldots \wedge x_n^{\varepsilon_n}$$

where

$$x^\varepsilon \text{ is a replacement for } \begin{cases} x & \text{if } \varepsilon = 1, \\ \neg x & \text{if } \varepsilon = 0. \end{cases}$$

(If f is identically equal to 0 then $f(x_1,\ldots,x_n) = x_1 \wedge \neg x_1$.) Therefore, $\{\wedge,\vee,\neg\}$ is a *complete* set of Boolean operations in the sense that every Boolean function can be "obtained from $\{\wedge,\vee,\neg\}$" using superpositions and usual manipulation of variables.

Are there any other complete sets of Boolean functions? Well, clearly $\{\neg,\wedge\}$, $\{\neg,\vee\}$ and $\{\neg,\Rightarrow\}$ are complete.

It is usually tedious but easy to show that a certain set is complete. But how can one show that some set of Boolean functions is *not* complete?

Example 4.1 (a) $\{\wedge,\vee\}$ is not complete: since $0\wedge 0 = 0\vee 0 = 0$, every function f that can be obtained from these two functions also fulfills $f(0,\ldots,0) = 0$. Therefore, $\neg$ cannot be obtained from $\wedge$ and $\vee$.

(b) $\{\Rightarrow\}$ is not complete: since $1 \Rightarrow 1 = 1$, every function f that can be obtained from this function has the property that $f(1,\ldots,1) = 1$. Therefore, $\neg$ cannot be obtained from $\Rightarrow$.

(c) $\{\neg,\Leftrightarrow\}$ is not complete. To see this, note that $\neg x = 1 + x$ and $x \Leftrightarrow y = 1 + x + y$, where $+$ is the addition in $GF(2)$. Now it is easy to show that if a function f can be obtained from $\neg$ and $\Leftrightarrow$ then $f(x_1,\ldots,x_n) = b + a_1x_1 + \ldots + a_nx_n$ for some $b, a_1, \ldots, a_n \in \{0,1\}$. On the other hand, $x \wedge y$ is not of this form, so $\{\neg,\Leftrightarrow\}$ is not complete.

Problem 4.2 (a) *Make precise the meaning of the phrase "a function can be 'obtained' from a set of functions".*

(b) *Given a set of functions F, decide whether F is complete.*

Our major reference for general clone theory is [48], and we rely on [61] for the applications of clone theory in universal algebra. The section on abstract clones follows the idea presented by B. Csákány in his addendum to the Hungarian translation of "A course in universal algebra" by S. Burris and H. P. Sankappanavar [15]. Most unreferenced statements in this text can be found (in this or a similar form) in one of the three books.

4.1 Clones

Throughout this text, A is a finite set with at least two elements. Let $\mathscr{O}_A^{(n)} = A^{A^n}$ be a set of all n-ary operations on A, $n \geqslant 1$, and $\mathscr{O}_A = \bigcup_{n\geqslant 1} \mathscr{O}_A^{(n)}$ be the set of all finitary

operations on A. The arity of an operation f is denoted by $\mathrm{ar}(f)$. For $F \subseteq \mathscr{O}_A$ let $F^{(n)} = F \cap \mathscr{O}_A^{(n)}$.

Let π_i^n denote the *i-th n-ary projection*:

$$\pi_i^n(x_1,\ldots,x_i,\ldots,x_n) = x_i$$

and let Π_A denote the set of all projections of all finite arities on A. Let $f \in \mathscr{O}_A^{(n)}$ and $g_1,\ldots,g_n \in \mathscr{O}_A^{(m)}$. The *superposition of f and $g_1,\ldots,g_n$* is an operation $h = f(g_1,\ldots,g_n)$ of arity m defined by

$$h(x_1,\ldots,x_m) = f(g_1(x_1,\ldots,x_m),\ldots,g_n(x_1,\ldots,x_m)).$$

Definition 4.3 A set $C \subseteq \mathscr{O}_A$ is called a *clone of operations on A* if

- $\Pi_A \subseteq C$, and
- for each $f,g_1,\ldots,g_n \in C$ such that $\mathrm{ar}(f) = n$ and $\mathrm{ar}(g_1) = \ldots = \mathrm{ar}(g_n)$, we have $f(g_1,\ldots,g_n) \in C$.

The legend says that the name *clone* came around 1936 from Marshall Hall (1910–1990) as a convenient abbreviation for the "closed one". The requirement that the clone contain projections makes it easy to formalise "usual manipulations with variables" as the following examples show.

Example 4.4 Let C be a clone and $f(x,y,z,u,v) \in C$. Then

$$f(y,y,y,x,x) \in C \quad \text{since} \quad f(y,y,y,x,x) = f(\pi_2^5,\pi_2^5,\pi_2^5,\pi_1^5,\pi_1^5)(x,y,z,u,v),$$
$$f(z,x,u,y,v) \in C \quad \text{since} \quad f(z,x,u,y,v) = f(\pi_3^5,\pi_1^5,\pi_4^5,\pi_2^5,\pi_5^5)(x,y,z,u,v).$$

It is obvious that the intersection of an arbitrary family of clones is a clone. So for an $F \subseteq \mathscr{O}_A$ let $\mathrm{Cln}(F)$ denote the least clone that contains F. This clone is said to be *generated* by F. A clone C is said to be *finitely generated* if $C = \mathrm{Cln}(\{f_1,\ldots,f_n\})$ for some $f_i \in \mathscr{O}_A$. Now it is easy to show:

Theorem 4.5 *Let $F \subseteq \mathscr{O}_A$ and let $\mathscr{F}$ be an algebraic type such that $\mathbf{A} = (A,F)$ is of type $\mathscr{F}$. Then $g \in \mathrm{Cln}(F)$ if and only if there is an $\mathscr{F}$-term t such that $g = t^{\mathbf{A}}$.*

Theorem 4.6 *All clones of operations on a finite set A form an algebraic lattice $\mathscr{L}_A$ under set inclusion. The least element of the lattice is Π_A, the greatest element is $\mathscr{O}_A$, and the lattice operations are given by:*

$$\bigwedge_\alpha C_\alpha = \bigcap_\alpha C_\alpha, \quad \text{and} \quad \bigvee_\alpha C_\alpha = \mathrm{Cln}(\bigcup_\alpha C_\alpha).$$

The last theorem tells us that it is possible to introduce and algebraic structure on $\mathscr{O}_A$ in such a way that $C \subseteq \mathscr{O}_A$ is a clone if and only if it is a subuniverse of the algebra. Presenting such an algebra explicitely is easy. Consider the following four operations on $\mathscr{O}_A$:

- for $f \in \mathscr{O}_A^{(m)}$ and $g \in \mathscr{O}_A^{(n)}$ let

$$(f * g)(x_1, \ldots, x_{m+n-1}) = f(g(x_1, \ldots, x_n), x_{n+1}, \ldots, x_{m+n-1});$$

- for $f \in \mathscr{O}_A^{(1)}$ let $\zeta f = \tau f = \triangle f = f$;
- for $f \in \mathscr{O}_A^{(n)}$, $n \geqslant 2$, let

$$\begin{aligned}(\zeta f)(x_1, x_2, \ldots, x_n) &= f(x_2, \ldots, x_n, x_1)\\ (\tau f)(x_1, x_2, x_3, \ldots, x_n) &= f(x_2, x_1, x_3, \ldots, x_n)\\ (\triangle f)(x_1, x_2, \ldots, x_{n-1}) &= f(x_1, x_1, x_2, \ldots, x_{n-1}).\end{aligned}$$

The algebra $(\mathscr{O}_A, *, \zeta, \tau, \triangle, \pi_1^2)$ is referred to as the *Mal'cev algebra*.

Theorem 4.7 *A set $C \subseteq \mathscr{O}_A$ is a clone if and only if it is a subuniverse of the Mal'cev algebra.*

Definition 4.8 A set $F \subseteq \mathscr{O}_A$ is *complete* if $\mathrm{Cln}(F) = \mathscr{O}_A$.

Now that we have firmly established the terminology, we can finally start looking for a completeness criterion. First, one can easily show that

Proposition 4.9 *A set $F \subseteq \mathscr{O}_A$ is complete if and only if $F \not\subseteq C$ for every clone $C \neq \mathscr{O}_A$.*

Needless to say that this criterion is pretty useless. We are going to turn it into a much more usefull criterion by focusing of some very *special* clones. The structure of the lattice of clones given in Theorem 4.6 suggests that the lattice of clones could be dual-atomic, which actually *is the case*. The key argument is the following statement due to Yablonskiĭ:

Theorem 4.10 (Yablonskiĭ 1958, [66]) *Let C be a clone on a finite set. Then C is finitely generated if and only if there exist maximal subclones of C, every proper subclone of C is contained in a maximal subclone of C and maximal subclones of C are finite in number.*

The proof of this theorem requires some preparation. Let $F \subseteq \mathscr{O}_A$ and $g \in \mathscr{O}_A^{(n)}$ for some integer $n \geqslant 1$. We say that *an operation g preserves operations from F* if $g(f_1,\ldots,f_n) \in F$ whenever $f_1,\ldots,f_n \in F$ are of the same arity.

Lemma 4.11 *Let $n \geqslant 1$ be an integer, let C be a clone and let $F \subset C^{(n)}$. There exists at most one maximal subclone D of C such that $D^{(n)} = F$.*

Proof. Let D be a maximal subclone of C such that $D^{(n)} = F$ and let us show that D has to be unique. Let $P_C(F)$ be the set of all operations in C that preserve operations from F and let us show that $D = P_C(F)$.

It is easy to see that $P_C(F)$ is a clone and that $D \subseteq P_C(F) \subseteq C$. Since D is a maximal subclone of C it follows that either $D = P_C(F)$ or $P_C(F) = C$. Suppose that $P_C(F) = C$ and let us show that this leads to a contradiction. Take any $f \in C^{(n)}$. Then $f \in C = P_C(F)$, so f preserves operations from F. According to the choice of D we have that $F = D^{(n)}$, so F contains $\pi_1^n, \ldots, \pi_n^n$. Now, f preserves operations from F, so $f(\pi_1^n,\ldots,\pi_n^n) \in F$, i.e. $f \in F$. This shows that $C^{(n)} \subseteq F$ – contradiction.

Therefore, $P_C(F) \neq C$ and the maximality of D now yields $P_C(F) = D$, which shows that D is uniquely determined by F. □

Let us now go back to the proof of Theorem 4.10.

Proof. (of Theorem 4.10) ($\Leftarrow$) Let $D_1, \ldots, D_k$ be maximal proper subclones of C. Take any $f_1 \in C \setminus D_1, \ldots, f_k \in C \setminus D_k$ and let $F = \{f_1,\ldots,f_k\}$. From $F \subseteq C$ it follows that $\mathrm{Cln}(F) \subseteq C$. Now, if $\mathrm{Cln}(F) \subset C$, then it is a proper subclone of C and, by assumption, it is contained in a maximal subclone of C, say D_i. But, this is not possible, since $f_i \in \mathrm{Cln}(F) \setminus D_i$. Therefore, $\mathrm{Cln}(F) = C$.

($\Rightarrow$) Assume that C is finitely generated, say, $C = \mathrm{Cln}(\{f_1,\ldots,f_k\})$. Since C is finitely generated, it follows immediately that the union of every chain of proper subclones of C is again a proper subclone of C, so by Zorn's Lemma, we get that maximal subclones of C exist and that every proper subclone of C is contained in a maximal subclone of C.

Let us now show that there are only finitely many maximal subclones of C. Let $n_i = \mathrm{ar}(f_i)$, $N = \max\{n_1,\ldots,n_k\}$, let $\mathscr{M}$ be the set of all maximal subclones of C and consider the mapping

$$\varphi : \mathscr{M} \to \mathscr{P}(\mathscr{O}_A^{(N)}) : D \mapsto D^{(N)}.$$

Since $\mathrm{Cln}(C^{(N)}) = C$ it follows that $D^{(N)} \subset C^{(N)}$ for every $D \in \mathscr{M}$. Lemma 4.11 now imples that φ is injective, so $|\mathscr{M}| \leqslant |\mathscr{P}(\mathscr{O}_A^{(N)})|$, and $\mathscr{P}(\mathscr{O}_A^{(N)})$ is a finite set due to the fact that A is finite. □

Theorem 4.12 *$\mathscr{O}_A$ is finitely generated.*

Proof. If $|A| = 2$ we have seen at the very beginning that $\mathscr{O}_A$ is finitely generated. For $|A| \geqslant 3$ we follow the straightforward idea of Werner and Wille. Fix two distinct elements from A and call them 0 and 1. For $a \in A$ let

$$c_a(x) = a, \quad \text{and} \quad \chi_a(x) = \begin{cases} 1, & x = a \\ 0, & x \neq a. \end{cases}$$

Choose binary operations $+, \cdot \in \mathscr{O}_A^{(2)}$ so that $0+x = x+0 = 0$, $x \cdot 0 = 0$ and $x \cdot 1 = x$ for all $x \in A$. Then it is easy to see that

$$f(x_1, \ldots, x_n) = \sum_{(a_1,\ldots,a_n) \in A^n} c_{f(a_1,\ldots,a_n)}(x_1) \cdot \chi_{a_1}(x_1) \cdot \chi_{a_2}(x_2) \cdot \ldots \cdot \chi_{a_n}(x_n).$$

Note that $+$ and $\cdot$ need not be associative, and that the representation is valid regardless of the actual order of taking sums and taking products.

Therefore, $\mathscr{O}_A$ can be generated by the following finite set of functions: $\{+, \cdot\} \cup \{c_a : a \in A\} \cup \{\chi_a : a \in A\}$. □

Corollary 4.13 *$\mathscr{O}_A$ has maximal subclones, they are finite in number, and every proper subclone of $\mathscr{O}_A$ is contained in one of the maximal subclones.*

Definition 4.14 Maximal subclones of $\mathscr{O}_A$ are called *maximal clones on A*.

Finally, we come to a much better completeness criterion, which tells us that in order to prove that a set F is complete, it suffices to check only finitely many clones. Nevertheless, the result needs some more refinements.

Theorem 4.15 *A set $F \subseteq \mathscr{O}_A$ is complete if and only if $F \not\subseteq C$ for every maximal clone C on A.*

We shall conclude this section by two important completeness criteria. The Słupecki completeness criterion says that all unary operations together with an essential operation (to be defined shortly) constitute a complete set. The theorem of Webb, on the other hand, shows that there exist one-element complete sets.

We say that an operation $f \in \mathscr{O}_A^{(n)}$ *depends on its i-th argument* if there exist $a, b, c_1, \ldots, c_{i-1}, c_{i+1}, \ldots, c_n \in A$ such that $a \neq b$ and

$$f(c_1, \ldots, c_{i-1}, a, c_{i+1}, \ldots, c_n) \neq f(c_1, \ldots, c_{i-1}, b, c_{i+1}, \ldots, c_n).$$

An operation f is *essential* if it is surjective and depends on at least two of its arguments. Let $\mathrm{im}(f) = f(A^n)$ denote the set of images of f.

Lemma 4.16 (The Main Lemma of Yablonskiĭ, 1958 [66]) *Assume that $f \in \mathscr{O}_A$ depends on at least two arguments and let $n = \mathrm{ar}(f)$.*

(a) If $|\mathrm{im}(f)| \geqslant 3$ then there exist $X_1, \ldots, X_n \subseteq A$ such that $|X_i| \leqslant 2$ for all i, and $|f(X_1, \ldots, X_n)| \geqslant 3$.

(b) If $|\mathrm{im}(f)| = k \geqslant 3$ then there exist $X_1, \ldots, X_n \subseteq A$ such that $|X_i| \leqslant k-1$ for all i, and $|f(X_1, \ldots, X_n)| = k$.

Proof. Without loss of generality we may assume that f depends on the first two arguments.

(a) Since f depends on the first argument, there exist $a, a', b_2, \ldots, b_n \in A$ such that

$$f(a, b_2, \ldots, b_n) = p$$
$$f(a', b_2, \ldots, b_n) = q \neq p.$$

Case 1: $f(a,A,\ldots,A) \neq \{p,q\}$. Since f depends on the second argument we have $|f(a,A,\ldots,A)| \geqslant 2$, so there exists an $r \in f(a,A,\ldots,A) \setminus \{p,q\}$. Choose $c_2, \ldots, c_n \in A$ so that

$$f(a, c_2, \ldots, c_n) = r.$$

Now let $X_1 = \{a, a'\}, X_2 = \{b_2, c_2\}, \ldots, X_n = \{b_n, c_n\}$. Clearly $p, q, r \in f(X_1, \ldots, X_n)$ so $|f(X_1, \ldots, X_n)| \geqslant 3$.

Case 2: $f(a,A,\ldots,A) = \{p,q\}$. From $|\mathrm{im}(f)| \geqslant 3$ we know that there is an $r \notin \{p,q\}$ and $c_1, \ldots, c_n \in A$ such that

$$f(c_1, c_2, \ldots, c_n) = r.$$

By the assumption, $f(a, c_2, \ldots, c_n) \in \{p,q\}$ so without loss of generality we may assume that

$$f(a, c_2, \ldots, c_n) = p.$$

But $f(a,A,\ldots,A) = \{p,q\}$ whence follows that there exist $d_2, \ldots, d_n \in A$ so that

$$f(a, d_2, \ldots, d_n) = q.$$

Now let $X_1 = \{a, c_1\}, X_2 = \{c_2, d_2\}, \ldots, X_n = \{c_n, d_n\}$. Clearly $p, q, r \in f(X_1, \ldots, X_n)$ so $|f(X_1, \ldots, X_n)| \geqslant 3$.

(b) Let $\mathrm{im}(f) = \{a_1, \ldots, a_k\}$. According to (a) there exist $Y_1, \ldots, Y_n \subseteq A$ such that $|Y_i| \leqslant 2$ and $|f(Y_1, \ldots, Y_n)| \geqslant 3$. Let $a_1, a_2, a_3 \in f(Y_1, \ldots, Y_n)$. For each $j \geqslant 4$ choose $b_1^j, \ldots, b_n^j \in A$ so that

$$f(b_1^j, \ldots, b_n^j) = a_j.$$

Put $X_i = Y_i \cup \{b_i^4, \ldots, b_i^k\}$, $i \in \{1, \ldots, n\}$. Then clearly $|X_i| \leqslant k-1$ for all i and $f(X_1, \ldots, X_n) = \mathrm{im}(f)$. □

Theorem 4.17 (Słupecki 1939, [57]) *Let $|A| \geqslant 3$ and let $F \subseteq \mathscr{O}_A$. If F contains an essential operation and if $\mathscr{O}_A^{(1)} \subseteq F$, then F is complete.*

Theorem 4.18 (Webb 1935, [64]) *Let $A = \{0, 1, \ldots, k-1\}$ and $x \uparrow y = \max(x, y) + 1$, where $+$ denotes addition mod k. Then $\{\uparrow\}$ is a complete set of operations.*

Proof. Let $s(x) = x \uparrow x = x + 1$. Then $\max(x, y) = s^{k-1}(x \uparrow y)$. Next, we can obtain constant maps as

$$c_a(x) = s^{a+1}(\max\{s^j(x) : j \in A\}),$$

and characteristic functions as

$$\chi_a(x) = s(\max\{s^j(x) : j \in A \text{ and } a + j \neq k-1\}) = \begin{cases} 0, & x \neq a \\ k-1, & x = a \end{cases}$$

Next,

$$\overline{x} = (k-1) - x = \max\{s^{k-j}(\max(\chi_j(x), j)) : j \in A\}$$

and

$$\min(x, y) = \overline{\max(\overline{x}, \overline{y})}.$$

The statement now follows by the same argument as in the proof of Theorem 4.12, where we let max play the role of $+$, min the role of $\cdot$ and $k-1$ the role of 1. □

Definition 4.19 A *Sheffer operation* is any operation $f \in \mathscr{O}_A$ such that $\{f\}$ is a complete set.

Sheffer operations were named after H. M. Sheffer who in 1913 discovered a Sheffer operation $x \uparrow y = \neg(x \wedge y)$ on $\{0, 1\}$ (see [56]). We see now that $\mathscr{O}_A$ has a Sheffer operation whenever A is a finite set with at least two elements.

4.2 Galois connections

We have seen that in order to show that a certain set of operations is complete it suffices to show how to produce operations from some other complete set of operations. But how does one show that a certain set F of functions is *not* complete? The idea is to find a property P which is preserved under superpositions, to show that every operation $f \in F$ has the property P, and to find an operation g which does not have the property. Then $g \notin \mathrm{Cln}(F)$ (since, by the choice of P, every operation in F has the property P) and F is not complete.

Example 4.20 (a) Let $A = \{1, \ldots, n\}$ and let $\min(x,y)$, $\max(x,y)$ denote the usual binary minimum and maximum operations on integers. To show that $\{\min, \max\}$ is not complete, if suffices to note that $\min(1,1) = \max(1,1) = 1$ and hence for every $f \in \mathrm{Cln}(\min, \max)$ we have $f(1, \ldots, 1) = 1$. On the other hand, the constant map $c_n(x) = n$ does not have this property.

(b) Let $A = \{0, 1, \ldots, p-1\}$ where p is an odd prime and let $+$ be the addition mod p. Let $\sim x = 1 + x$ and let us show that $\{\sim, +\}$ is not complete. For every $f \in \mathrm{Cln}(\sim, +)$ there are $a_1, \ldots, a_n, b \in A$ such that $f(x_1, \ldots, x_n) = a_1x_1 + a_2x_2 + \ldots + a_nx_n + b$ whence follows that $\min \notin \mathrm{Cln}(\sim, +)$.

The standard approach is to encode properties by finitary relations and then interpret "f has property P" by "f preserves (an appropriately chosen relation) ρ". Let $\mathscr{R}_A^{(h)} = \mathscr{P}(A^h)$ denote the set of all h-ary relations on A and let $\mathscr{R}_A = \bigcup_{h \geqslant 1} \mathscr{R}_A^{(h)}$ be the set of all finitary relations on A. If $\rho \in \mathscr{R}_A^{(h)}$ and $\rho \neq \varnothing$, we write $\mathrm{ar}(\rho) = h$. We take $\mathrm{ar}(\varnothing) = 0$.

Definition 4.21 An operation $f \in \mathscr{O}_A^{(n)}$ *preserves* a relation $\rho \in \mathscr{R}_A^{(h)}$ if

$$\begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{h1} \end{bmatrix} \in \rho, \begin{bmatrix} a_{12} \\ a_{22} \\ \vdots \\ a_{h2} \end{bmatrix} \in \rho, \ldots, \begin{bmatrix} a_{1n} \\ a_{2n} \\ \vdots \\ a_{hn} \end{bmatrix} \in \rho \text{ implies } \begin{bmatrix} f(a_{11}, a_{12}, \ldots, a_{1n}) \\ f(a_{21}, a_{22}, \ldots, a_{2n}) \\ \vdots \\ f(a_{h1}, a_{h2}, \ldots, a_{hn}) \end{bmatrix} \in \rho,$$

or, equivalently, if ρ is a subuniverse of $(A, f)^h$. We also say that ρ is an *invariant relation* of f.

Example 4.22 (a) Let $\leqslant$ be a partial order on A. Then "f preserves $\leqslant$" is equivalent to the fact that f is monotonous with respect to $\leqslant$.

(b) "f preserves $\{a\}$" is equivalent to $f(a, \ldots, a) = a$.

(c) If ε is an equivalence relation on A then "f preserves ε" is equivalent to ε being a congruence of (A, f).

Let O and R be nonempty sets and let $\rho \subseteq O \times R$ be a binary relation. Define $\overrightarrow{\rho} : \mathscr{P}(O) \to \mathscr{P}(R)$ and $\overleftarrow{\rho} : \mathscr{P}(R) \to \mathscr{P}(O)$ by

$$\overrightarrow{\rho}(F) = \{r \in R : \forall f \in F\ (f \rho r)\} \quad \text{and} \quad \overleftarrow{\rho}(Q) = \{f \in O : \forall r \in Q\ (f \rho r)\},$$

where $F \subseteq O$ and $Q \subseteq R$. Then the pair $(\overrightarrow{\rho}, \overleftarrow{\rho})$ is called the *Galois connection* between $\mathscr{P}(O)$ and $\mathscr{P}(R)$ with respect to ρ. The proof of the following theorem can be found e.g. in [40]:

Theorem 4.23 *Let $(\overrightarrow{\rho}, \overleftarrow{\rho})$ be a Galois connection between $\mathscr{P}(O)$ and $\mathscr{P}(R)$ with respect to ρ, and let $F, F_1, F_2 \in \mathscr{P}(O)$ and $Q, Q_1, Q_2 \in \mathscr{P}(R)$.*

(1) If $F_1 \subseteq F_2$ then $\overrightarrow{\rho}(F_1) \supseteq \overrightarrow{\rho}(F_2)$.
If $Q_1 \subseteq Q_2$ then $\overleftarrow{\rho}(Q_1) \supseteq \overleftarrow{\rho}(Q_2)$.

(2) $F \subseteq \overleftarrow{\rho}(\overrightarrow{\rho}(F))$ and $Q \subseteq \overrightarrow{\rho}(\overleftarrow{\rho}(Q))$.

(3) $\overrightarrow{\rho}(F) = \overrightarrow{\rho}(\overleftarrow{\rho}(\overrightarrow{\rho}(F)))$ and $\overleftarrow{\rho}(Q) = \overleftarrow{\rho}(\overrightarrow{\rho}(\overleftarrow{\rho}(Q)))$.

(4) $L_O = \{\overleftarrow{\rho}(Q) : Q \subseteq R\}$ and $L_R = \{\overrightarrow{\rho}(F) : F \subseteq O\}$ are dually isomorphic complete lattices with respect to $\subseteq$. The dual isomorphisms are

$$\overrightarrow{\rho} : L_O \to L_R \quad \text{and} \quad \overleftarrow{\rho} : L_R \to L_O.$$

The relation "... preserves..." generates a Galois connection between operations and relations and the corresponding operators $\overrightarrow{\rho}$ and $\overleftarrow{\rho}$ are commonly denoted by $\operatorname{Pol} Q$ and $\operatorname{Inv} F$, $Q \subseteq \mathscr{R}_A$, $F \subseteq \mathscr{O}_A$:

$$\operatorname{Pol} Q = \{f \in \mathscr{O}_A : f \text{ preserves every } \rho \in Q\}$$
$$\operatorname{Inv} F = \{\rho \in \mathscr{R}_A : \text{ every } f \in F \text{ preserves } \rho\}.$$

It is easy to show that $\operatorname{Pol} Q$ is a clone of operations for every Q. But the converse is also true. Before we move on to the proof, let us introduce some more notation. Let $\mathbf{x}_1, \ldots, \mathbf{x}_n$ be tuples understood as column-vectors. Then $f(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ is a column-vector obtained by applying f to the rows of the matrix $[\mathbf{x}_1 \ \ldots \mathbf{x}_n]$, i.e.

$$\text{if} \quad \mathbf{x}_1 = \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{h1} \end{bmatrix}, \ldots, \mathbf{x}_n = \begin{bmatrix} a_{1n} \\ a_{2n} \\ \vdots \\ a_{hn} \end{bmatrix} \quad \text{then} \quad f(\mathbf{x}_1, \ldots \mathbf{x}_n) = \begin{bmatrix} f(a_{11}, a_{12}, \ldots, a_{1n}) \\ f(a_{21}, a_{22}, \ldots, a_{2n}) \\ \vdots \\ f(a_{h1}, a_{h2}, \ldots, a_{hn}) \end{bmatrix}.$$

$$\begin{array}{ccccc} a & a & a & \dots & a \\ a & a & a & \dots & b \\ \vdots & \vdots & \vdots & & \vdots \\ z & z & z & \dots & y \\ z & z & z & \dots & z \end{array} \quad \rightsquigarrow \quad \begin{array}{ccccc} \begin{bmatrix} a \\ a \\ \vdots \\ z \\ z \end{bmatrix} & \begin{bmatrix} a \\ a \\ \vdots \\ z \\ z \end{bmatrix} & \begin{bmatrix} a \\ a \\ \vdots \\ z \\ z \end{bmatrix} & \dots & \begin{bmatrix} a \\ b \\ \vdots \\ y \\ z \end{bmatrix} \\ \uparrow & \uparrow & \uparrow & \dots & \uparrow \\ \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 & \dots & \mathbf{x}_n \end{array}$$

Figure 4.1: The special relation from the proof of Theorem 4.24

Theorem 4.24 (Bondarčuk, Kalužnin, Kotov, Romov 1969, [8]) *Let $C \subseteq \mathscr{O}_A$. The following statements are equivalent:*

(1) C is a clone.

(2) $C = \operatorname{Pol} Q$ for some $Q \subseteq \mathscr{O}_A$.

(3) $C = \operatorname{Pol}\operatorname{Inv} C$.

Proof. $(3) \Rightarrow (2) \Rightarrow (1)$ is easy. In order to show $(1) \Rightarrow (3)$ it suffices to show $\operatorname{Pol}\operatorname{Inv} C \subseteq C$ since the other inclusion is true for any Galois connection. Take any $g \in \operatorname{Pol}\operatorname{Inv} C$ and let $n = \operatorname{ar}(g)$. We shall now construct a special relation of arity $|A|^n$. List all n-tuples from A^n in lexicographic order, denote the column-vectors by $\mathbf{x}_1, \dots, \mathbf{x}_n$, Fig. 4.1, and let

$$\theta_n(C) = \{ f(\mathbf{x}_1, \dots, \mathbf{x}_n) : f \in C \}.$$

Since C is a clone, it is easy to show that $\theta_n(C) \in \operatorname{Inv} C$, so g preserves $\theta_n(C)$. Now, $\mathbf{x}_1, \dots, \mathbf{x}_n \in \theta_n(C)$ since $\pi^n_1, \dots, \pi^n_n \in C$, so from the fact that g preserves $\theta_n(C)$ it follows that $g(\mathbf{x}_1, \dots, \mathbf{x}_n) \in \theta_n(C)$. Therefore, there is an $f \in C$ such that $g(\mathbf{x}_1, \dots, \mathbf{x}_n) = f(\mathbf{x}_1, \dots, \mathbf{x}_n)$ and from the construction of $\mathbf{x}_1, \dots, \mathbf{x}_n$ it follows $g = f \in C$. □

Corollary 4.25 *Let $F \subseteq \mathscr{O}_A$.*

(a) $\operatorname{Cln}(F) = \operatorname{Pol}\operatorname{Inv} F$.

(b) If every $f \in F$ preserves a relation ρ then every $f \in \operatorname{Cln}(F)$ preserves ρ.

Galois closed sets on the relational side are somewhat more complicated. It is possible to define closed sets of relations using "trivial" relations and a requirement that the set be closed with respect to certain "superpositions", but this approach is less usual. Later on, we shall treat relational clones in a more standard way, as subuniverses of a certain algebra on $\mathscr{R}_A$.

We start with the trivial relations. Let $n \geqslant 1$ and let ε be an equivalence relation on $\{1,\dots,n\}$. Then *the n-ary ε-diagonal* is the following relation:

$$\delta_n^\varepsilon = \{(x_1,\dots,x_n) \in A^n : \forall i,j\ ((i,j) \in \varepsilon \Rightarrow x_i = x_j)\}.$$

Let $\delta_0 = \varnothing$ be the *zero-ary diagonal* and let Δ_A denote the set of all diagonals on all arities $\geqslant 0$.

Example 4.26 Let $12|3$ denote the equivalence relation on $\{1,2,3\}$ with two blocks, $\{1,2\}$ and $\{3\}$. Then $\delta_3^{12|3} = \{(x,x,y) : x,y \in A\}$.

Let $\tau,\rho,\sigma \in \mathscr{R}_A$ be relations such that $\mathrm{ar}(\tau) = \mathrm{ar}(\rho) + \mathrm{ar}(\sigma)$, let t be an integer and $f : \{1,\dots,t\} \to \{1,\dots,m+n\}$ be a mapping where $m = \mathrm{ar}(\rho)$ and $n = \mathrm{ar}(\sigma)$. We define the *f-superposition of τ with ρ and σ* denoted by $[\tau,\rho,\sigma]_f$ as follows: $[\tau,\rho,\varnothing]_f = [\tau,\varnothing,\sigma]_f = [\varnothing,\varnothing,\varnothing]_\varnothing = \varnothing$ and in case $m \geqslant 1$, $n \geqslant 1$ we let

$$\begin{aligned}[\tau,\rho,\sigma]_f = \{(x_{f(1)},x_{f(2)},\dots,x_{f(t)}) \in A^t : (x_1,x_2,\dots,x_{m+n}) \in \tau,\\ (x_1,\dots,x_m) \in \rho \text{ and } (x_{m+1},\dots,x_{m+n}) \in \sigma\}.\end{aligned}$$

Definition 4.27 A *relational clone on A* is any set $Q \subseteq \mathscr{R}_A$ such that

- $\Delta_A \subseteq Q$, and
- If $\tau,\rho,\sigma \in \mathscr{R}_A$ such that $\mathrm{ar}(\tau) = \mathrm{ar}(\rho) + \mathrm{ar}(\sigma)$, then for every $f : \{1,\dots,t\} \to \{1,\dots,m+n\}$, where $m = \mathrm{ar}(\rho)$ and $n = \mathrm{ar}(\sigma)$, we have $[\tau,\rho,\sigma]_f \in Q$.

It is easy to see that the intersection of an arbitrary nonempty family of relational clones is again a relational clone. Therefore, for every $Q \subseteq \mathscr{R}_A$ there exists the least relational clone that contains Q. We say that this *relational clone is generated by Q* and denote it by $\mathrm{Clr}(Q)$.

In order to present more conventional descriptions of relational clones, we have to introduce several operations on relations:

- for $\rho \in \mathscr{R}_A^{(m)}$ and $\sigma \in \mathscr{R}_A^{(n)}$ we define the *relational product* $\times$ and the *relational composition* $\circ$ by

$$\begin{aligned}\rho \times \sigma &= \{(x_1,\dots,x_m,y_1,\dots,y_n) : (x_1,\dots,x_m) \in \rho \text{ and } (y_1,\dots,y_n) \in \sigma\}\\ \rho \circ \sigma &= \{(x_1,\dots,x_{m-1},y_2,\dots,y_n) : \exists z \in A\ ((x_1,\dots,x_{m-1},z) \in \rho \text{ and}\\ &\qquad (z,y_2,\dots,y_n) \in \sigma)\};\end{aligned}$$

- for $\rho \in \mathscr{R}_A^{(1)}$ or $\rho = \varnothing$ let $\zeta\rho = \tau\rho = \triangle\rho = \rho$; for $\rho \in \mathscr{R}_A^{(n)}$, $n \geqslant 2$, let

$$\begin{aligned}\zeta\rho &= \{(x_1,x_2,\ldots,x_n) : (x_2,\ldots,x_n,x_1) \in \rho\},\\ \tau\rho &= \{(x_1,x_2,x_3,\ldots,x_n) : (x_2,x_1,x_3,\ldots,x_n) \in \rho\},\\ \triangle\rho &= \{(x_1,x_2,\ldots,x_{n-1}) : (x_1,x_1,x_2,\ldots,x_{n-1}) \in \rho\},\end{aligned}$$

  be the *cyclic permutation of variables, transposition of the first two variables* and the *identification of the first two variables*; and

- for $\rho \in \mathscr{R}_A^{(n)}$ with $n \geqslant 2$ and $i,j \in \{1,\ldots,n\}$ such that $i < j$ let

$$\triangle_{i,j}(\rho) = \{(x_1,\ldots,x_i,\ldots,x_{j-1},x_{j+1},\ldots,x_n) : (x_1,\ldots,x_n) \in \rho \text{ and } x_i = x_j\}$$

  denote the variation of $\triangle$ that operates on arbitrary two coordinates;

- for $\rho \in \mathscr{R}_A^{(n)}$ and any mapping $f : \{1,\ldots,m\} \to \{1,\ldots,n\}$, let

$$\mathrm{pr}_f(\rho) = \{(x_{f(1)},x_{f(2)},\ldots,x_{f(m)}) : (x_1,\ldots,x_n) \in \rho\}.$$

  denote the *f-projection* of ρ. We shal often write simply $\mathrm{pr}_{i_1\ldots i_k}$ e.g.

$$\mathrm{pr}_{522}(\rho) = \{(x_5,x_2,x_2) : (x_1,x_2,x_3,x_4,x_5) \in \rho\} \text{ where } f = \begin{pmatrix} 1 & 2 & 3 \\ 5 & 2 & 2 \end{pmatrix}.$$

Theorem 4.28 *Let $Q \subseteq \mathscr{R}_A$. The following are equivalent:*

(1) Q is a relational clone;

(2) $\Delta_A \subseteq Q$ and Q is closed with respect to intersection of relations of the same arity, relational products and f-projections; and

(3) Q is a subuniverse of the algebra $(\mathscr{R}_A, \circ, \zeta, \tau, \triangle, \delta_3^{12|3})$.

Proof. (1) $\Leftrightarrow$ (2): Note that $[\tau,\rho,\sigma]_f = \mathrm{pr}_f(\tau \cap (\rho \times \sigma))$. On the other hand, $\mathrm{pr}_f(\rho) = [A^{n+1},\rho,A]_{f'}$, where $n = \mathrm{ar}(\rho)$ and $f' : \{1,\ldots,t\} \to \{1,\ldots,n+1\}$ is given by $f'(i) = f(i)$; $\rho \times \sigma = [A^{n+m},\rho,\sigma]_{\mathrm{id}}$, where $n = \mathrm{ar}(\rho)$ and $m = \mathrm{ar}(\sigma)$; and $\rho \cap \sigma = [\rho \times A, \sigma, A]_{f'}$, where $n = \mathrm{ar}(\rho)$ and $f' : \{1,\ldots,n\} \to \{1,\ldots,n+1\}$ is given by $f'(i) = i$.

(2) $\Rightarrow$ (3): Unary operations are eazy: $\zeta(\rho) = \mathrm{pr}_{n12\ldots n-1}(\rho)$, $\tau(\rho) = \mathrm{pr}_{213\ldots n}(\rho)$ and $\triangle(\rho) = \mathrm{pr}_{23\ldots n}(\rho \cap \delta_n^{12|3|\ldots|n})$. As for $\circ$ let $n = \mathrm{ar}(\rho)$, $m = \mathrm{ar}(\sigma)$ and let ε be the equivalence relation on $\{1,\ldots,m+n\}$ whose only nontrivial block is $\{n,n+1\}$. Then $\rho \circ \sigma = \mathrm{pr}_{1\ldots n-1,n+2\ldots n+m}(\delta_{n+m}^{\varepsilon} \cap (\rho \times \sigma))$.

(3) $\Rightarrow$ (2): Note first that compositions of ζ and τ can achieve any permutation of variables. Now $\delta_3^{12|3} \circ \zeta\zeta(\delta_3^{12|3}) = \delta_4^{12|34}$, $\triangle\zeta^3(\delta_4^{12|34}) = \delta_3^{123}$ and $\triangle\triangle(\delta_3^{12|3}) =$

$\delta^1_1 = A$. Then $\rho \times \sigma = \rho \circ \delta^{12|34}_4 \circ \sigma$, $A \circ \rho$ omits the first variable of ρ, while $\delta^{123}_3 \circ \rho$ doubles the first variable of ρ. It is easy but rather technical to show that using the last three operations we can get all the diagonals and all the f-projections. Next we note that each $\triangle_{i,j}$ can be obtained from ζ, τ and $\triangle$. Finally, if ρ and σ are relations of arity n then $\rho \cap \sigma = \triangle_{n,n+1}\triangle_{n-1,n+1}\ldots\triangle_{2,n+1}\triangle_{1,n+1}(\rho \times \sigma)$ which completes the proof. □

The proof of the following theorem is due to Bondarčuk, Kalužnin, Kotov and Romov (1969) and independently Geiger (1968).

Theorem 4.29 (Geiger 1968, Bondarčuk, Kalužnin, Kotov, Romov 1969, [25, 8]) *Let $Q \subseteq \mathscr{R}_A$. The following statements are equivalent:*

(1) Q is a relational clone.

(2) $Q = \operatorname{Inv} F$ for some $F \subseteq \mathscr{R}_A$.

(3) $Q = \operatorname{Inv}\operatorname{Pol} Q$.

Proof. (3) $\Rightarrow$ (2) $\Rightarrow$ (1) is easy. Let us show (1) $\Rightarrow$ (3). Clearly $Q \subseteq \operatorname{Inv}\operatorname{Pol} Q$ since this is true for every Galois connection. Let us show that $\operatorname{Inv}\operatorname{Pol} Q \subseteq Q$. Let $C = \operatorname{Pol} Q$ and let us first show that $\theta_n(C) \in Q$ for all $n \geqslant 1$, where $\theta_n(C)$ is the relation defined in the proof of Theorem 4.24. Recall also the vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$ from the definition of $\theta_n(C)$, let $q = |A|^n$ and let

$$\gamma_n = \bigcap\{\eta \in Q : \operatorname{ar}(\eta) = q \text{ and } \eta \supseteq \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}\}.$$

It is obvious that $\gamma_n \in Q$ since Q is closed with respect to finite intersections. We are going to show that $\theta_n(C) = \gamma_n$ whence $\theta_n(C) \in Q$ follows immediately.

First, $\theta_n(C) \subseteq \gamma_n$ since $\mathbf{x}_1, \ldots, x_n \in \gamma_n$ and γ_n is preserved by every $f \in C$ ($C = \operatorname{Pol} Q$, so $\operatorname{Inv} C = \operatorname{Inv}\operatorname{Pol} Q \supseteq Q \ni \gamma_n$). Assume that $\theta_n(C) \subset \gamma_n$, take any $\mathbf{r} = (u_1, \ldots, u_q) \in \gamma_n \setminus \theta_n(C)$ and consider $g_{\mathbf{r}}$ defined by $g_{\mathbf{r}}(\mathbf{x}_1, \ldots, \mathbf{x}_n) = \mathbf{r}$. From $\mathbf{r} \notin \theta_n(C)$ it follows that $g_{\mathbf{r}} \notin C = \operatorname{Pol} Q$, so there is a $\rho \in Q$ such that $g_{\mathbf{r}}$ does not preserve ρ. Therefore, there exist $\mathbf{y}_1, \ldots, \mathbf{y}_n \in \rho$ such that $g_{\mathbf{r}}(\mathbf{y}_1, \ldots, \mathbf{y}_n) \notin \rho$. Let $m = \operatorname{ar}(\rho)$ and consider the tuples $\mathbf{x}_i \times \mathbf{y}_i$ of length $q+m$ (obtained by concatenating $\mathbf{x}_i$ and $\mathbf{y}_i$) as columns of an $(q+m) \times n$-matrix, Fig. 4.2. Let X be the top $q \times n$-submatrix formed by $\mathbf{x}_1, \ldots, \mathbf{x}_n$ and let Y be the bottom $m \times n$-submatrix formed by $\mathbf{y}_1, \ldots, \mathbf{y}_n$. Since the rows of X are *all posssible* n-tuples of elements from A and rows of Y are *some* n-tuples of elements of Y, every row of Y appears as a row in X. Assume that the j-th row of Y appears as the h_j-th row of X. Let ε be the equivalence relation on $\{1, \ldots, q+m\}$ generated by the pairs $(h_j, q+j)$, $j \in \{1, \ldots, m\}$, and let

$$\rho' = \operatorname{pr}_{1\ldots q}(\delta^{\varepsilon}_{q+m} \cap (\gamma_n \times \rho)).$$

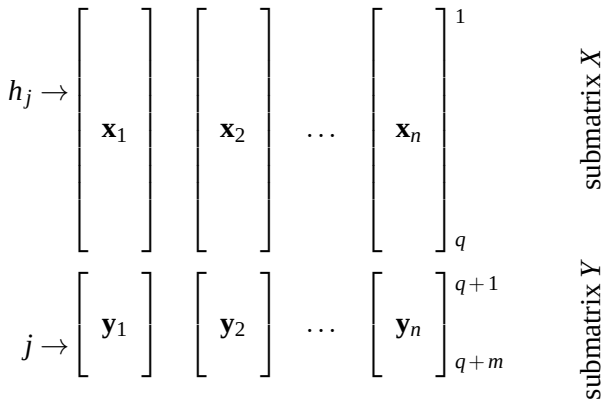


Figure 4.2: The proof of Theorem 4.29

Clearly, $\rho' = [\delta^{\varepsilon}_{q+m}, \gamma_n, \rho]_{1\dots q} \in Q$ and $\rho' \subseteq \gamma_n$. On the other hand, from $\mathbf{x}_i \times \mathbf{y}_i \in \delta^{\varepsilon}_{q+m} \cap (\gamma_n \times \rho)$ it follows that $\mathbf{x}_1, \dots, \mathbf{x}_n \in \rho'$.

Let us show that $\mathbf{r} \notin \rho'$. Since $g_{\mathbf{r}}(\mathbf{x}_1, \dots, \mathbf{x}_n) = (u_1, \dots, u_q)$ and since the j-th row of Y is equal to the h_j-th row of X it follows that $g_{\mathbf{r}}(\mathbf{y}_1, \dots, \mathbf{y}_n) = (u_{h_1}, \dots, u_{h_m})$ and by the choice of $\mathbf{y}_1, \dots, \mathbf{y}_n$ we have $(u_{h_1}, \dots, u_{h_m}) \notin \rho$. Now, if $\mathbf{r} \in \rho'$ then $(u_1, \dots, u_q, u_{h_1}, \dots, u_{h_m}) \in \gamma_n \times \rho$, whence follows $(u_{h_1}, \dots, u_{h_m}) \in \rho$, which is impossible. So, $\mathbf{r} \notin \rho'$.

Therefore, $\mathbf{r} \in \gamma_n \setminus \rho'$, whence $\rho' \subset \gamma_n$. But this is impossible since γ_n is constructed as the least relation in Q that contains $\mathbf{x}_1, \dots, \mathbf{x}_n$. This shows $\gamma_n = \theta_n(C)$ and thus $\theta_n(C) \in Q$.

Finally, let us show that $\operatorname{Inv}\operatorname{Pol} Q \subseteq Q$. Take any $\sigma \in \operatorname{Inv}\operatorname{Pol} Q$, let $n = |\sigma|$ and $t = \operatorname{ar}(\sigma)$. Write all tuples in σ as column vectors and denote this matrix by M. The rows of matrix X (Fig. 4.2) are all possible n-tuples of elements of A, so there are indices $i_1, \dots, i_t \in \{1, \dots, q\}$ such that the k-th row of M is equal to the i_k-th row of X, $k \in \{1, \dots, t\}$, so $\sigma = \operatorname{pr}_{i_1 \dots i_t}(\{\mathbf{x}_1, \dots, \mathbf{x}_n\})$. Let us show that $\sigma = \operatorname{pr}_{i_1 \dots i_t}(\theta_n(C))$. Since $\{\mathbf{x}_1, \dots, \mathbf{x}_n\} \subseteq \theta_n(C)$ it immeditaly follows that $\sigma = \operatorname{pr}_{i_1 \dots i_t}(\{\mathbf{x}_1, \dots, \mathbf{x}_n\}) \subseteq \operatorname{pr}_{i_1 \dots i_t}(\theta_n(C))$. To show the other inclusion, take any $(v_1, \dots, v_t) \in \operatorname{pr}_{i_1 \dots i_t}(\theta_n(C))$. Then there is a $\mathbf{z} = (z_1, \dots, z_q) \in \theta_n(C)$ such that $z_{i_1} = v_1, \dots, z_{i_t} = v_t$. But $\mathbf{z} = f(\mathbf{x}_1, \dots, \mathbf{x}_n)$ for some $f \in C$ so $z_{i_j} = f(x_{1i_j}, x_{2i_j}, \dots, x_{ni_j})$, where $\mathbf{x}_j = (x_{j1}, \dots, x_{jq})$. Since $\sigma = pr_{i_1 \dots i_t}(\{\mathbf{x}_1, \dots, \mathbf{x}_n\})$ it follows that $(x_{1i_1}, \dots, x_{1i_t}), \dots, (x_{ni_1}, \dots, x_{ni_t}) \in \sigma$. Now $f \in C$ and $\sigma \in \operatorname{Inv} C$ imply that f preserves σ, so $(v_1, \dots, v_t) = (z_{i_1}, \dots, z_{i_t}) \in \sigma$.

Therefore, $\sigma = \operatorname{pr}_{i_1 \dots i_t}(\theta_n(C))$. Since $\theta_n(C) \in Q$ we have $\sigma = \operatorname{pr}_{i_1 \dots i_t}(\theta_n(C)) \in Q$. This completes the proof that $\operatorname{Inv}\operatorname{Pol} Q = Q$. □

Corollary 4.30 $\operatorname{Clr}(Q) = \operatorname{Inv}\operatorname{Pol} Q$.

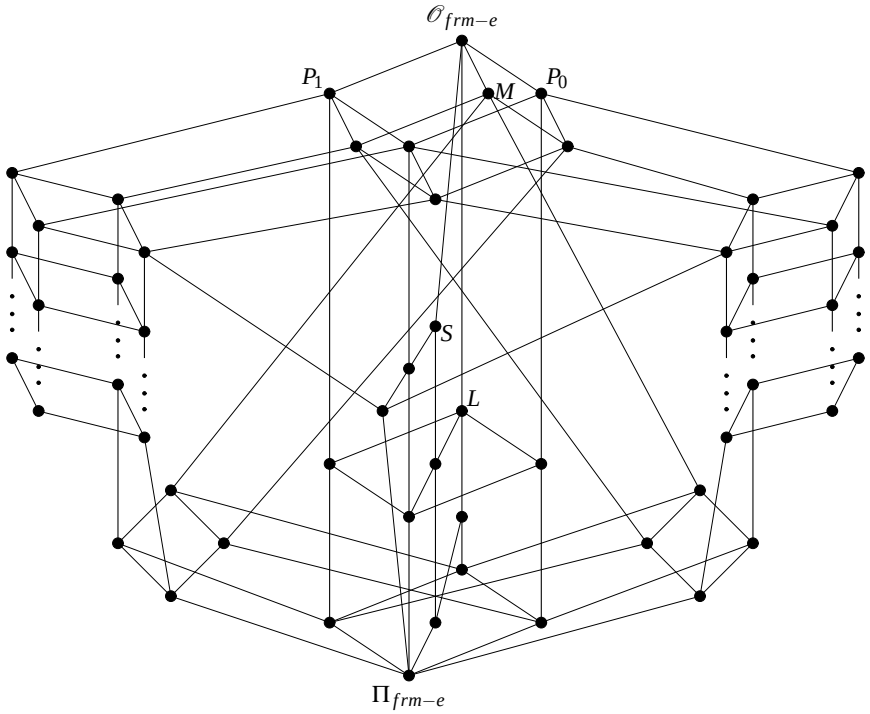


Figure 4.3: The lattice of clones on $\{0,1\}$ (E. Post)

4.3 On the number of clones

Emil Post classified all possible clones on $\{0,1\}$, and hence in a natural sense all possible 2-valued propositional logics. His work was first presented in 1920 as an addendum to his Ph. D. Thesis, and it was finally published in the book *Two-valued Iterative Systems of Mathematical Logic*, Princeton, 1941 (see [49]). The classification of E. Post was presented in a more modern notation by R. Lyndon in [35]. The lattice of clones on a two-element set is given in Fig. 4.3. Among other things, it explicitely shows that there are countably many clones of operations on A if $|A| = 2$.

Yu. I. Yanov and A. A. Muchnik showed in 1959 that in case of a finite set with $|A| \geqslant 3$ the number of clones is continuum.

Theorem 4.31 (Yanov, Muchnik 1959, [67]) *Let A be a finite set with at least three elements. Then the number of clones on A is continuum.*

Proof. Since clones are special subsets of the countably infinite set $\mathscr{O}_A$ it follows immediately that $|\mathscr{L}_A| \leqslant c$. To show that we have an equality, it suffices to construct

a family of clones with continuum elements.

Let $0,1,2 \in A$ be three distinct elements of A and for each $n \geqslant 3$ define $f_n \in \mathscr{O}_A^{(n)}$ and $\rho_n, \sigma_n \in \mathscr{R}_A^{(n)}$ by

$$\sigma_n = \{(1,2,2,\ldots,2,2),(2,1,2,\ldots,2,2),\ldots,(2,2,2,\ldots,2,1)\},$$

$$f_n(x_1,\ldots,x_n) = \begin{cases} 1, & (x_1,\ldots,x_n) \in \sigma_n \\ 0, & \text{otherwise,} \end{cases}$$

$$\rho_n = \sigma_n \cup \{(x_1,\ldots,x_n) \in \{0,1,2\}^n : \exists i\ (x_i = 0)\}.$$

Let us first show that $f_n \notin \mathrm{Pol}\{\rho_n\}$ and that $f_k \in \mathrm{Pol}\{\rho_n\}$ whenever $k \neq n$. The first part of the claim is easy:

$$\begin{array}{cccccccc} 1 & 2 & 2 & \ldots & 2 & 2 & \xrightarrow{f_n} & 1 \\ 2 & 1 & 2 & \ldots & 2 & 2 & \xrightarrow{f_n} & 1 \\ 2 & 2 & 1 & \ldots & 2 & 2 & \xrightarrow{f_n} & 1 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots \\ 2 & 2 & 2 & \ldots & 1 & 2 & \xrightarrow{f_n} & 1 \\ 2 & 2 & 2 & \ldots & 2 & 1 & \xrightarrow{f_n} & 1 \\ \in & \in & \in & \ldots & \in & \in & & \notin \\ d & d & d & & d & d & & d \end{array}$$

As for the second part of the claim, assume that $k < n$ and take any $\mathbf{x}_1, \ldots, \mathbf{x}_k \in \rho_n$. If at least one of the $\mathbf{x}_i$'s has a zero coordinate, then $f_k(\mathbf{x}_1, \ldots, \mathbf{x}_k)$ also has a zero coordinate and hence $f_k(\mathbf{x}_1, \ldots, \mathbf{x}_k) \in \rho_n$. If, however, $\mathbf{x}_1, \ldots, \mathbf{x}_k \in \sigma_n$ then at least one of the rows will consist of 2's and $f_k(\mathbf{x}_1, \ldots, \mathbf{x}_k)$ will again have a zero coordinate, ensuring $f_k(\mathbf{x}_1, \ldots, \mathbf{x}_k) \in \rho_n$. E.g., if $k = 3$ and $n = 4$ this situation may be illustrated by

$$\begin{array}{cccccc} 1 & 2 & 2 & \xrightarrow{f_3} & 1 \\ 2 & 1 & 2 & \xrightarrow{f_3} & 1 \\ 2 & 2 & 1 & \xrightarrow{f_3} & 1 \\ 2 & 2 & 2 & \xrightarrow{f_3} & 0 \\ \| & \| & \| & & \in \\ \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 & & d \end{array}$$

Finally, let $k > n$ and take any $\mathbf{x}_1, \ldots, \mathbf{x}_k \in \rho_n$. It at least one of the $\mathbf{x}_i$'s has a zero coordinate, then as in case $k < n$ we conclude that $f_k(\mathbf{x}_1, \ldots, \mathbf{x}_k)$ also has a zero coordinate and hence $f_k(\mathbf{x}_1, \ldots, \mathbf{x}_k) \in \rho_n$. If, however, $\mathbf{x}_1, \ldots, \mathbf{x}_k \in \sigma_n$ then at least

one of the rows will contain at least two 1's and $f_k(\mathbf{x}_1, \ldots, \mathbf{x}_k)$ will again have a zero coordinate, ensuring $f_k(\mathbf{x}_1, \ldots, \mathbf{x}_k) \in \rho_n$. E.g., if $k = 5$ and $n = 4$ this situation may be illustrated by

$$\begin{array}{ccccccc} 1 & 2 & 2 & 2 & 2 & \xrightarrow{f_5} & 1 \\ 2 & 1 & 2 & 2 & 2 & \xrightarrow{f_5} & 1 \\ 2 & 2 & 1 & 2 & 1 & \xrightarrow{f_5} & 0 \\ 2 & 2 & 2 & 1 & 2 & \xrightarrow{f_5} & 1 \\ \| & \| & \| & \| & \| & & \in \rho \\ \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 & \mathbf{x}_4 & \mathbf{x}_5 & & \end{array}$$

We are now ready to complete the proof. Let $H = \{f_3, f_4, \ldots\}$ be the set of functions we have just constructed and define $\varphi : \mathscr{P}(H) \to \mathscr{L}_A$ by $\varphi(F) = \mathrm{Cln}(F)$. Let us show that φ is injective. Suppose $F_1 \neq F_2$ for some $F_1, F_2 \subseteq H$, but $\mathrm{Cln}(F_1) = \mathrm{Cln}(F_2)$. Since $F_1 \neq F_2$, there is an $n \geqslant 3$ such that $f_n \in F_1 \setminus F_2$ (or the other way around). So, $f_n \in F_1 \subseteq \mathrm{Cln}(F_1) = \mathrm{Cln}(F_2)$. Since $f_n \notin F_2$ we have that every $g \in F_2$ preserves ρ_n. Therefore, every $g \in \mathrm{Cln}(F_2)$ preserves ρ_n and consequently $f_n \in \mathrm{Cln}(F_2)$ preserves ρ_n – contradiction.

This shows that φ is injective and hence $|\mathscr{L}_A| \geqslant |\mathscr{P}(H)| = c$. □

4.4 Minimal clones

A clone C is called a *minimal clone* if $C \neq \Pi_A$ and $D \subseteq C$ implies $D = \Pi_A$ or $D = C$ for every clone D. Minimal clones are atoms in $\mathscr{L}_A$ and it is easy to see that every minimal clone is of the form $\mathrm{Cln}(f)$ for some $f \in \mathscr{O}_A \setminus \Pi_A$.

We are going to show that on a finte set every clone $\neq \Pi_A$ contains a minimal clone and that there are finitely many minimal clones. We also show that all minimal clones split into five types. We start by introducing some terminology. A *polymer* of an operation f is every operation that can be obtained from f by identifying certain variables. Note that f is *not* a polymer of itself: in order to obtain a polymer one *has to* identify at least two variables. For example, $g(x,y) = f(x,y,x,y)$ is a polymer of f. A ternary operation f on A is called a *majority operation* if

$$f(a,a,b) = f(a,b,a) = f(b,a,a) = a,$$

and a *minority operation* if

$$f(a,a,b) = f(a,b,a) = f(b,a,a) = b,$$

for all $a, b \in A$. For $n \geqslant 3$ and $k \in \{1, \ldots, n\}$, an n-ary operation f is called a *k-th n-ary semiprojection* if it is not a projection, but

$$f(a_1, \ldots, a_n) = a_k$$

whenever $a_1,\dots,a_n \in A$ have the property that $|\{a_1,\dots,a_n\}| < n$.

Lemma 4.32 (Świerczkowski 1960, [58]) *Let f be an n-ary operation on A and $n \geqslant 4$. Then f is a semiprojection if and only if every polymer of f is a projection.*

Proof. ($\Rightarrow$) Obvious.

($\Leftarrow$) First let us note that

$$f(x_1,y,y,x_4,\dots,x_n) \notin \{\pi_2^{n-1},\pi_3^{n-1}\} \quad \text{or} \quad f(y,x_2,x_3,y,x_5,\dots,x_n) \notin \{\pi_1^{n-1},\pi_4^{n-1}\}.$$

(Suppose to the contrary that, e.g., $f(x_1,y,y,x_4,\dots,x_n) = \pi_2^{n-1}$ and $f(y,x_2,x_3,y,x_5,\dots,x_n) = \pi_1^{n-1}$. Then $f(z,y,y,z,x_5,\dots,x_n)$ would at the same time have to be the first and the second projection, which is impossible.) So, without loss of generality we may assume that $f(x_1,y,y,x_4,\dots,x_n) = \pi_1^{n-1}$, i.e. $f(x_1,y,y,x_3,\dots,x_n) = x_1$. Now take $i,j \in \{2,\dots,n\}$ such that $i < j$ and consider

$$f(x_1,\dots,x_{i-1},y,x_{i+1},\dots,x_{j-1},y,x_{j+1},\dots,x_n). \tag{4.1}$$

We know that it is a projection and since $f(x_1,y,y,\dots,y) = x_1$, the polymer (4.1) has to be the first projection. Using this fact one now easily shows that

$$f(y,x_2,\dots,x_{i-1},y,x_{i+1},\dots,x_n)$$

is again the first projection, for all $i \in \{2,\dots,n\}$. Therefore, f is a first semiprojection. □

Theorem 4.33 (Rosenberg 1986, [52]) *Every clone $C \neq \Pi_A$ contains an operation which belongs to one of the following five classes of operations:*

(1) a nonidentical unary operation;

(2) a binary idempotent operation which is not a projection;

(3) a majority operation;

(4) a minority operation;

(5) a semiprojection.

Proof. Take an operation $f \in C \setminus \Pi_A$ of the least possible arity and let $n = \mathrm{ar}(f)$. If $n = 1$ we have case (1). Otherwise, if $n \geqslant 2$ the choice of f implies that all polymers of f have to be projections. So, if $n = 2$ we have case (2), while in case of $n \geqslant 4$ Lemma 4.32 yields that f is a semiprojection (case (5)).

Finally, let $n = 3$. Since all polymers of f are projections, $f(x,x,y)$, $f(x,y,x)$ and $f(y,x,x)$ are either π^2_1 or π^2_2, so we have eight cases to consider. Five cases are straightforward:

$$\begin{array}{llll} f(x,x,y)=x, & f(x,y,x)=x, & f(y,x,x)=x: & \text{case (3)} \\ f(x,x,y)=y, & f(x,y,x)=y, & f(y,x,x)=y: & \text{case (4)} \\ f(x,x,y)=x, & f(x,y,x)=x, & f(y,x,x)=y: & \text{a first semiprojection} \\ f(x,x,y)=x, & f(x,y,x)=y, & f(y,x,x)=x: & \text{a second semiprojection} \\ f(x,x,y)=y, & f(x,y,x)=x, & f(y,x,x)=x: & \text{a third semiprojection} \end{array}$$

In the remaining three cases (possibly after a permutation of variables) one obtains an operation that satisfies

$$f(x,x,y) = f(y,x,y) = f(y,x,x) = y$$

but then $g(x,y,z) = f(x, f(x,y,z), z)$ is a majority operation (case (3)). □

Corollary 4.34 *For every minimal clone C we have that $C = \mathrm{Cln}(f)$, where f is an operation that belongs to one of the five classes of operations listed in Theorem 4.33 and minimal clones are finite in number. Moreover, every clone $C \neq \Pi_A$ contains a minimal clone,*

Proof. Let C be a minimal clone and let $f \in C$ be an operation whose existence is guaranteed by Theorem 4.33. Then $\mathrm{Cln}(f) \subseteq C$ and since $f \notin \Pi_A$, the minimality of C yields $\mathrm{Cln}(f) = C$.

To show that there are finitely many minimal clones on A it suffices to show that each of the classes (1)–(5) in Theorem 4.33 is finite. Classes (1)–(4) are obviously finite. As for class (5), note that if f is a semiprojection then $\mathrm{ar}(f) \leqslant |A|$ (a semiprojection of arity $> |A|$ would have to be a projection, which by definition is not a semiprojection).

For an arbitrary clone $C \neq \Pi_A$ take $f \in C$ as in Theorem 4.33 and let us define a sequence of clones and operations as follows: $C_0 = C$, $f_0 = f$, and

- if $\mathrm{Cln}(f_i)$ is a minimal clone, let $C_{i+1} = C_i$ and $f_{i+1} = f_i$;
- if $\mathrm{Cln}(f_i)$ is not a minimal clone, choose C_{i+1} such that $\Pi_A \neq C_{i+1} \subset \mathrm{Cln}(f_i)$ and take $f_{i+1} \in C_{i+1}$ as in Theorem 4.33.

Then

- either $C_0 \supseteq \mathrm{Cln}(f_0) \supset \mathrm{Cln}(f_1) \supset \ldots \supset \mathrm{Cln}(f_k) = \mathrm{Cln}(f_{k+1}) = \ldots$ for some k,
- or $C_0 \supseteq \mathrm{Cln}(f_0) \supset \mathrm{Cln}(f_1) \supset \ldots \supset \mathrm{Cln}(f_i) \supset \ldots$

In the former case $\mathrm{Cln}(f_k)$ is a minimal clone contained in $C_0 = C$, so in order to somplete the proof it suffices to show that the latter case cannot occur. Suppose that $\mathrm{Cln}(f_i) \subset \mathrm{Cln}(f_j)$ whenever $i < j$. Then $f_i \neq f_j$ for all $i \neq j$ whence follows that there are infinitely many operations that belong to the classes (1)–(5) in Theorem 4.33. But we have shown in the previous paragraph that each of the five classes of operations is finite. Contradiction. □

The results presented in Theorem 4.33 and Corollary 4.34 have been a part of folklore for quite some time, and can be traced back to [17]. There are minimal clones that belong to each of the five types, as the following example shows.

Example 4.35 (1) Take an idempotent unary operation f which is not identity. Then $\mathrm{Cln}(f)$ is a minimal clone.

(2) Let $\wedge$ be a semilattice operation on A. Then $\mathrm{Cln}(\wedge)$ is a minimal clone.

(3) $\mathrm{Cln}(d)$ is a minimal clone if d is a dual discriminator on A, i.e.,

$$d(x,y,z) = \begin{cases} x, & x = y \\ z, & \text{otherwise.} \end{cases}$$

(4) If $(A,+)$ is an abelian group of exponent 2 and $f(x,y,z) = x+y+z$ then $\mathrm{Cln}(f)$ is a minimal clone.

(5) Let $n = |A| \geqslant 3$. Then $\mathrm{Cln}(l_n)$ is a minimal clone, where l_n is an n-ary operation defined by:

$$l_n(x_1,\ldots,x_n) = \begin{cases} x_n, & \{x_1,\ldots,x_n\} = A \\ x_1, & \text{otherwise.} \end{cases}$$

Theorem 4.36 (Rosenberg 1986, [52]) *Let C be a minimal clone generated by a unary operation f. Then $f \neq \mathrm{id}_A$ and either $f^2 = f$ or there is a prime number p such that $f^p = \mathrm{id}_A$.*

Let C be a minimal clone generated by a minority operation f. Then there is an abelian group $(A,+)$ of exponent 2 such that $f(x,y,z) = x+y+z$.

4.5 Maximal clones

We already know that there are finitely many maximal clones on a finite set and that every proper subclone of $\mathscr{O}_A$ is contained in a maximal clone. In this section we address one of the deepest and most influential results of clone theory, I. Rosenberg's classification of maximal clones. We start with a special case.

Theorem 4.37 (Yablonskiĭ 1958, [66]) *Let $|A| = k$ and let $\iota_A = \{(x_1,\dots,x_k) \in A^k : \exists i \neq j\ (x_i = x_j)\}$. Then $\mathrm{Pol}\{\iota_A\}$ is the set of all nonessential operations. It is a maximal clone, and it is the only maximal clone that contains $\mathscr{O}_A^{(1)}$.*

Proof. A nonessential operations is either essentially unary, of nonsurjective. It easily seen that both essentially unary and nonsurjective operations preserve ι_A whence follows that $\mathrm{Pol}\{\iota_A\}$ contains all nonessential operations. Let us show that every operation in $\mathrm{Pol}\{\iota_A\}$ is nonessential.

Suppose to the contrary that $\mathrm{Pol}\{\iota_A\}$ contains an essential operation f and let $A = \{a_1,\dots,a_k\}$. Then according to the Main Lemma of Yablonskiĭ 4.16 there exist $X_1,\dots,X_n \subseteq A$ such that $|X_i| \leqslant k-1$ for all i and $f(X_1,\dots,X_n) = A$. Therefore, for each $i \in \{1,\dots,k\}$ there exist $x_i^1 \in X_1$, …, $x_i^n \in X_n$ such that

$$f(x_i^1,\dots,x_i^n) = a_i.$$

For $\mathbf{x}^1 = (x_1^1,\dots,x_k^1)$, …, $\mathbf{x}^n = (x_1^n,\dots,x_k^n)$ we have $\mathbf{x}^1,\dots,\mathbf{x}^n \in \iota_A$ since $\{x_1^i,\dots,x_k^i\} \subseteq X_i$ and $|X_i| \leqslant k-1$. On the other hand, $f(\mathbf{x}^1,\dots,\mathbf{x}^n) = (a_1,\dots,a_k) \notin \iota_A$, so f does not preserve ι_A. This completes the proof that $\mathrm{Pol}\{\iota_A\}$ is the set of all nonessential operations.

It is now very easy to show that $\mathrm{Pol}\{\iota_A\}$ is a maximal clone. Take any $f \in \mathscr{O}_A \setminus \mathrm{Pol}\{\iota_A\}$. Then f is an essential operation and $\mathrm{Cln}(\{f\} \cup \mathrm{Pol}\{\iota_A\}) = \mathscr{O}_A$ by the Słupecki completeness criterion (Theorem 4.17). The clone $\mathrm{Pol}\{\iota_A\}$ clearly contains all unary maps, and Lemma 4.11 ensures that no other maximal clone contains $\mathscr{O}_A^{(1)}$. □

For the general case, let us first show that each maximal clone is completely determined by a single relation.

Proposition 4.38 (Kuznecov 1961, [32]) *(a) If $\mathrm{Pol}\, Q = \mathscr{O}_A$ then $Q \subseteq \Delta_A$.*

(b) A clone $C \neq \mathscr{O}_A$ is a maximal clone if and only if $C = \mathrm{Pol}\{\rho\}$ for every $\rho \in \mathrm{Inv}C \setminus \Delta_A$.

Proof. *(a)* Take any $\rho \in Q$ and let $n = ar(\rho)$. Without loss of generality we may assume that there are no systematically repeated coordinates in ρ, i.e.,

$$\neg\, \exists i,j\ (i \neq j \text{ and } \forall (x_1,\dots,x_n) \in \rho\ (x_i = x_j))$$

for otherwise we can safely remove systematically repeated coordinates to obtain ρ' with the property $\mathrm{Pol}\{\rho\} = \mathrm{Pol}\{\rho'\}$. Therefore, for every $i,j \in \{1,\dots,n\}$ such that $i < j$ there exists an $\mathbf{x}^{ij} = (x_1^{ij},\dots,x_n^{ij}) \in \rho$ such that $x_i^{ij} \neq x_j^{ij}$. Take arbitrary

$a_1,\dots,a_n \in A$ and any map $f \in \mathscr{O}_A$ of arity $\binom{n}{2}$ which satisfies the following:

$$\begin{aligned} f(x_1^{12},x_1^{13},\dots,x_1^{n,n-1}) &= a_1 \\ f(x_2^{12},x_2^{13},\dots,x_2^{n,n-1}) &= a_2 \\ &\vdots \\ f(x_n^{12},x_n^{13},\dots,x_n^{n,n-1}) &= a_n \end{aligned}$$

Note that this requirement makes sense because each pair of $\binom{n}{2}$-tuples in the list above differs at at least one place. Since $\mathrm{Pol}\{\rho\} = \mathscr{O}_A$, this f preserves ρ so from $\mathbf{x}^{ij} \in \rho$ for all i, j it follows that $(a_1,\dots,a_n) \in \rho$. But $(a_1,\dots,a_n)$ was arbitrary, so $\rho = A^n \in \Delta_A$.

(b) First note that $\mathrm{Inv}C$ contains a nondiagonal relation whenever $C \neq \mathscr{O}_A$, since otherwise one has $\mathrm{Inv}C \subseteq \Delta_A$ whence $C = \mathrm{Pol}\,\mathrm{Inv}C \supseteq \mathrm{Pol}\,\Delta_A = \mathscr{O}_A$, which contradicts the fact that C is a proper subclone of $\mathscr{O}_A$.

$(\Rightarrow)$ Let C be a maximal clone and take any $\rho \in \mathrm{Inv}C \setminus \Delta_A$. Then $\mathrm{Pol}\{\rho\} \supseteq \mathrm{Pol}\,\mathrm{Inv}C = C$. Since (a) implies that $\mathrm{Pol}\{\rho\} \neq \mathscr{O}_A$, maximality of C yields $\mathrm{Pol}\{\rho\} = C$.

$(\Leftarrow)$ Suppose $C \neq \mathscr{O}_A$ is not a maximal clone. Then there is a maximal clone M such that $C \subset M$. Take any $\rho \in \mathrm{Inv}M \setminus \Delta_A$. Then $\rho \in \mathrm{Inv}C$ since $\mathrm{Inv}C \supseteq \mathrm{Inv}M \ni \rho$, but $C \neq \mathrm{Pol}\{\rho\}$ since $C \subset M = \mathrm{Pol}\{\rho\}$. □

In particular, each maximal clone M takes the form $\mathrm{Pol}\{\rho\}$ for a nondiagonal relation ρ. One of the most influential results in clone theory is the explicite characterization of the maximal clones, obtained in 1970 by I. G. Rosenberg as the culmination of the work of many mathematicians. It is usually stated in terms of the following six classes of finitary relations on A (the so-called *Rosenberg relations*). For an $f \in \mathscr{O}_A^{(n)}$, let $f^\bullet$ denote the $(n+1)$-ary relation on A called the *graph of* f:

$$f^\bullet = \{(x_1,\dots,x_n, f(x_1,\dots,x_n)) : x_1,\dots,x_n \in A\}.$$

(R1) *Bounded partial orders.* These are partial orders on A with a least and a greatest element.

(R2) *Nontrivial equivalence relations.* These are equivalence relations on A distinct from $\delta_2^{12} = \{(x,x) : x \in A\}$ and A^2.

(R3) *Permutational relations.* These are relations of the form $\alpha^\bullet$ where α is a fixpoint-free permutation of A with all cycles of the same prime length p.

(R4) *Affine relations.* An *affine relation* is a relation of the form $f^\bullet$ where $f(x,y,z) = x-y+z$ for an elementary abelian p-group $(A,+)$ on A.

(R5) *Central relations.* All unary relations are central relations. For central relations of arity $h \geqslant 2$ the definition is as follows. The relation ρ is said to be *totally symmetric* if $(x_1,\ldots,x_h) \in \rho$ implies $(x_{\pi(1)},\ldots,x_{\pi(h)}) \in \rho$ for all permutations π, and it is said to be *totally reflexive* if $(x_1,\ldots,x_h) \in \rho$ whenever there are $i \neq j$ such that $x_i = x_j$. A $c \in A$ is *central* if $(c,x_2,\ldots,x_h) \in \rho$ for all $x_2,\ldots,x_h \in A$. Finally, $\rho \neq A^h$ is called *central* if it is totally reflexive, totally symmetric and has a central element.

(R6) *Regular relations.* Let $\Theta = \{\theta_1,\ldots,\theta_m\}$ be a family of equivalence relations. We say that Θ is an *h-regular family* if every θ_i has precisely h blocks, and additionally, if B_i is an arbitrary block of θ_i, $i \in \{1,\ldots,m\}$, then $\bigcap_{i=1}^m B_i \neq \varnothing$.

An h-ary relation $\rho \neq A^h$ is said to be *h-regular* if $h \geqslant 3$ and there is an h-regular family Θ such that $(x_1,\ldots,x_h) \in \rho$ if and only if for all $\theta \in \Theta$ there are distinct i, j with $(x_i,x_j) \in \theta$.

Theorem 4.39 (Rosenberg 1970, [51]) *A clone M is maximal if and only if there is a relation ρ from one of the classes (R1)–(R6) such that $M = \mathrm{Pol}\{\rho\}$.*

The proof of Rosenberg's theorem is very complicated. The shortest known proof comes from Quackenbush [50] and it is still rather complicated. The sketch of the proof that we are going to present follows the track of the Quackenbush's proof and relies on two nontrivial facts which we shall not prove. Recall that a finite algebra (A,F) is called *quasiprimal* if $\mathrm{Cln}(F) = \mathrm{Pol}\,Q$ where $Q = \{h^\bullet : h$ is an isomorphism between subalgebras of $(A,F)\}$. recall also that a finite algebra (A,F) is quasiprimal if and only if $\mathrm{Cln}(F)$ contains a discriminator (Pixley [45]).

Proposition 4.40 (a) *(Quackenbush [50]) If $F \not\subseteq \mathrm{Pol}\{\rho\}$ for every Rosenberg relation ρ, then $\mathrm{Cln}(F)$ contains a Mal'cev operation.*

(b) *(McKenzie) If (A,F) is a simple finite algebra which has no proper subalgebras and which has a Mal'cev term operation, then (A,F) is quasiprimal, or there is an elementary abelian p-group $(A,+)$ with the following property: for every $f \in F$ there are an $a \in A$ and endomorphisms ε_i of $(A,+)$ such that $f(x_1,\ldots,x_n) = \varepsilon_1(x_1) + \ldots + \varepsilon_n(x_n) + a$.*

Proof. (*of Rosenberg's theorem*)

($\Leftarrow$) This direction is somewhat easier. We have to show that $\mathrm{Pol}\{\rho\}$ is a maximal clone for every Rosenberg relation ρ. We shall demonstrate main ideas in case of relations from (R1).

The strategy will be as follows. Let ρ be a Rosenberg relation and let $C = \mathrm{Pol}\{\rho\}$. According to Proposition 4.38 (a) we obtain $C \neq \mathscr{O}_A$, so by the statement (b) of the same proposition it suffices to show that for every $\sigma \in \mathrm{Inv}C \setminus \Delta_A$ we

have $C = \mathrm{Pol}\{\sigma\}$. And this is equivalent to showing that for every $\sigma \in \mathrm{Inv}C \setminus \Delta_A$ we have $\rho \in \mathrm{Clr}(\sigma)$. (To see this, note that from $C = \mathrm{Pol}\{\sigma\}$ it follows that $\mathrm{Inv}C = \mathrm{Inv}\,\mathrm{Pol}\{\sigma\} = \mathrm{Clr}(\sigma)$ and $\rho \in \mathrm{Inv}C$ now yields $\rho \in \mathrm{Clr}(\sigma)$; conversely, if $\rho \in \mathrm{Clr}(\sigma)$ then $C = \mathrm{Pol}\{\rho\} \supseteq \mathrm{Pol}\{\sigma\}$, while from $\sigma \in \mathrm{Inv}C$ it follows that $\mathrm{Pol}\{\sigma\} \supseteq C$.)

Let $\leqslant$ be a bounded partial order on A with the least element 0 and the greatest element 1. Let $C = \mathrm{Pol}\{\leqslant\}$, take any $\sigma \in \mathrm{Inv}C \setminus \Delta_A$ and let us show that $\leqslant \in \mathrm{Clr}(\sigma)$. Without loss of generality we may assume that there are no systematically repeated coordinates in σ, i.e.

$$\neg\, \exists i, j\ (i \neq j \text{ and } \forall (x_1, \ldots, x_n) \in \sigma\ (x_i = x_j)).$$

Suppose first that there exist $s \neq t$ such that $\mathrm{pr}_{st}(\sigma) \subseteq \leqslant$. Let us show that $\mathrm{pr}_{st}(\sigma) = \leqslant$. Let $a \leqslant b$, take any $(p_1, \ldots, p_n) \in \sigma$ such that $p_s < p_t$ and define f by

$$f(x) = \begin{cases} a, & x \leqslant p_s \\ b, & \text{otherwise.} \end{cases}$$

Clearly $f \in \mathrm{Pol}\{\leqslant\}$, so f preserves σ as well. Therefore $(f(p_1), \ldots, f(p_n)) \in \sigma$ and hence $(a, b) \in \mathrm{pr}_{st}(\sigma)$. This shows $\leqslant = \mathrm{pr}_{st}(\sigma) \in \mathrm{Clr}(\sigma)$.

Suppose now that for every $s \neq t$ we have $\mathrm{pr}_{st}(\sigma) \not\subseteq \leqslant$ and let us show that this leads to $\sigma = A^n$. Let $|\sigma| = m$ and denote the elements of σ by $(p_{i1}, \ldots, p_{in})$, $i \in \{1, \ldots, m\}$. For every $j \in \{1, \ldots, n\}$ let $\mathbf{x}_j = (p_{1j}, \ldots, p_{mj})$, Fig. 4.4. Take arbitrary $(q_1, \ldots, q_n) \in A^n$ and define f by

$$f(y_1, \ldots, y_m) = \begin{cases} q_i, & (y_1, \ldots, y_m) = \mathbf{x}_i \\ 0, & \text{there exists a } j \text{ such that } (y_1, \ldots, y_m) \leqslant \mathbf{x}_j \\ 1, & \text{otherwise.} \end{cases}$$

The assumption $\mathrm{pr}_{st}(\sigma) \not\subseteq \leqslant$ for all $s \neq t$ means that all $\mathbf{x}_j$'s are incomparable, so $f \in \mathrm{Pol}\{\leqslant\}$. Therefore, f preserves σ as well, whence $(q_1, \ldots, q_n) \in \sigma$. But $(q_1, \ldots, q_n)$ was chosen arbitrarily, so $\sigma = A^n$.

$(\Rightarrow)$ We now know that $\mathrm{Pol}\{\rho\}$ is a maximal clone for every Rosenberg relation ρ. Suppose there is a maximal clone C which is *not* of the form $\mathrm{Pol}\{\rho\}$ for a Rosenberg relation ρ. Then $C \not\subseteq \mathrm{Pol}\{\rho\}$ for every Rosenberg relation ρ, so by Proposition 4.40 (a), C contains a Mal'cev operation. In particular, since $C \not\subseteq \mathrm{Pol}\{\rho\}$ where ρ is an equivalence relation it follows that (A, C) is simple, while $C \not\subseteq \mathrm{Pol}\{\rho\}$ where ρ is a central relation implies (A, C) has no proper subalgebras. Then by Proposition 4.40 (b) it follows that (A, C) is quasiprimal, or that there is an elementary abelian p-group $(A, +)$ such that for every $f \in F$ there are an

$$\begin{array}{ccccccc}
\mathbf{x}_1 & = & p_{11} & p_{21} & \dots & p_{m1} & \stackrel{f}{\to} & q_1 \\
\mathbf{x}_2 & = & p_{12} & p_{22} & \dots & p_{m2} & \stackrel{f}{\to} & q_2 \\
\vdots & & \vdots & \vdots & & \vdots & \vdots & \vdots \\
\mathbf{x}_n & = & p_{1n} & p_{2n} & \dots & p_{mn} & \stackrel{f}{\to} & q_n \\
 & & \cap & \cap & \dots & \cap & & \\
 & & \mathfrak{a} & \mathfrak{a} & & \mathfrak{a} & &
\end{array}$$

Figure 4.4: The proof that Pol$\{\leqslant\}$ is a maximal clone

$a \in A$ and endomorphisms ε_i of $(A,+)$ and $f(x_1,\dots,x_n) = \varepsilon_1(x_1)+\dots+\varepsilon_n(x_n)+a$. The latter possibility would imply $C \subseteq \mathrm{Pol}\{(x-y+z)^\bullet\}$ which contradicts the fact that $C \not\subseteq \mathrm{Pol}\{\rho\}$ for every Rosenberg relation ρ. Therefore, (A,C) is quasiprimal.

We are going to show that (A,C) is primal, by showing that $\mathrm{Aut}(A,C) = \{\mathrm{id}\}$ (Proposition 4.40 (c)). Take any $\varphi \in \mathrm{Aut}(A,C)$. Let B be the set of all fixpoints of φ. Since B is a subalgebra of (A,C) and (A,C) has no proper subalgebras, we get $B = \varnothing$ or $B = A$. Therefore, $\varphi = \mathrm{id}$ or φ has no fixpoints. Assume $\varphi \neq \mathrm{id}$. Then φ has no fixpoints. Let k be the length of the shortest cycle of φ and let

$$(a_{11}\dots a_{1k}),\dots,(a_{n1}\dots a_{nk})$$

be all the shortest cycles of φ. Then $\{a_{11},\dots,a_{1k},\dots,a_{n1},\dots,a_{nk}\}$ is the set of all the fixpoints of $\varphi^k \in \mathrm{Aut}(A,C)$ and hence $\{a_{11},\dots,a_{1k},\dots,a_{n1},\dots,a_{nk}\} = A$. From $k > 1$ it follows that there is a prime p such that $k = pm$ for some m. But then $\psi = \varphi^m$ is a fixpoint free automorphism of (A,C) of order p, and hence $C \subseteq \mathrm{Pol}\{\psi^\bullet\}$ – a contradiction.

Therefore, (A,C) is primal, i.e., $C = \mathscr{O}_A$, which contradicts the fact that C is a maximal clone. □

4.6 Describing clones by relations of bounded arity

Our next goal is to characterize clones uniquely determined by their invariant relations of arity at most k. Let $g \in \mathscr{O}_A^{(n)}$, $F \subseteq \mathscr{O}_A$ and let k be a positive integer. Suppose that for every $S \subseteq A^n$ such that $|S| = k$ there is an $f \in F^{(n)}$ such that $g|_S = f|_S$. Then we say that *g can be k-approximated by F*. Let $\mathrm{Loc}_k(F)$ denote the set of all $g \in \mathscr{O}_A$ that can be k-approximated by F. We say that a clone C is *k-locally closed* if $C = \mathrm{Loc}_k(C)$.

Lemma 4.41 *For every clone C and every $k \geqslant 1$, $\mathrm{Loc}_k(C) = \mathrm{Pol}\,\mathrm{Inv}^{(k)} C$.*

Proof. ($\subseteq$) Take any $g \in \mathrm{Loc}_k(C)$ and let us show that $g \in \mathrm{Pol}\,\mathrm{Inv}^{(k)} C$. Let $n = \mathrm{ar}(g)$. Take any $\rho \in \mathrm{Inv}^{(k)} C$ and any $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \rho$. Let $\mathbf{x}_i = (x_{i1}, \ldots, x_{ik})$, $i \in \{1, \ldots, n\}$. Form new n-tuples $\mathbf{y}_j = (x_{1j}, \ldots, x_{nj})$, $j \in \{1, \ldots, k\}$, and let $S = \{\mathbf{y}_1, \ldots, \mathbf{y}_k\}$.

$$\begin{array}{ccccccc} \mathbf{y}_1 & \to & x_{11} & x_{21} & \ldots & x_{n1} \\ \mathbf{y}_2 & \to & x_{12} & x_{22} & \ldots & x_{n2} \\ \vdots & & \vdots & \vdots & & \vdots \\ \mathbf{y}_k & \to & x_{1k} & x_{2k} & \ldots & x_{nk} \\ & & \uparrow & \uparrow & \ldots & \uparrow \\ & & \mathbf{x}_1 & \mathbf{x}_2 & \ldots & \mathbf{x}_n \end{array}$$

Since $g \in \mathrm{Loc}_k(C)$ there is an $f \in C^{(n)}$ such that $g|_S = f|_S$, i.e., $g(\mathbf{x}_1, \ldots, \mathbf{x}_n) = f(\mathbf{x}_1, \ldots, \mathbf{x}_n)$. Now, $f \in C$ and $\rho \in \mathrm{Inv}^{(k)} C$ means that $f(\mathbf{x}_1, \ldots, \mathbf{x}_n) \in \rho$. Therefore, $g(\mathbf{x}_1, \ldots, \mathbf{x}_n) \in \rho$. This shows that g preserves ρ, and since ρ was arbitrary, we get $g \in \mathrm{Pol}\,\mathrm{Inv}^{(k)} C$.

($\supseteq$) Take any $g \in \mathrm{Pol}\,\mathrm{Inv}^{(k)} C$ and let $\mathrm{ar}(g) = n$. Let $S = \{\mathbf{y}_1, \ldots, \mathbf{y}_k\} \subseteq A^n$ be any k-element subset of A^n and let us show that there exists an $f \in C^{(n)}$ such that $g|_S = f|_S$. Let $\mathbf{y}_j = (x_{1j}, \ldots, x_{nj})$, $j \in \{1, \ldots, k\}$, and form the tuples $\mathbf{x}_i = (x_{i1}, \ldots, x_{ik})$, $i \in \{1, \ldots, n\}$, as in the diagram above. Finally, let

$$\theta = \{ f(\mathbf{x}_1, \ldots, \mathbf{x}_n) : f \in C^{(n)} \}.$$

Clearly, $\theta \in \mathrm{Inv}^{(k)} C$, so g preserves θ. Therefore, $g(\mathbf{x}_1, \ldots, \mathbf{x}_n) \in \theta$, whence follows that there is an $f \in C^{(n)}$ such that $g(\mathbf{x}_1, \ldots, \mathbf{x}_n) = f(\mathbf{x}_1, \ldots, \mathbf{x}_n)$. But this means that $g|_S = f|_S$ and thus $g \in \mathrm{Loc}_k(C)$. □

Theorem 4.42 (Szábo 1978, Pöschel 1979 [59, 46]) *Let C be a clone and let k be a positive integer. Then the following are equivalent:*

(1) C is k-locally closed;

(2) $C = \mathrm{Pol}\, Q$ for some $Q \subseteq \mathscr{R}_A^{(k)}$;

(3) $C = \mathrm{Pol}\,\mathrm{Inv}^{(k)} C$.

Proof. (3) $\Rightarrow$ (2) is trivial.

(2) $\Rightarrow$ (1): Let $C = \mathrm{Pol}\, Q$ for some $Q \subseteq \mathscr{R}_A^{(k)}$. To show that C is k-locally closed it suffices to show that $\mathrm{Loc}_k(C) \subseteq C$, for the other inclusion is trivial. But the proof that $\mathrm{Loc}_k(C) \subseteq \mathrm{Pol}\, Q$ is analogous to the proof of inclusion ($\subseteq$) in Lemma 4.41.

(1) $\Rightarrow$ (3): $C = \mathrm{Loc}_k(C)$ since C is k-locally closed, and $\mathrm{Loc}_k(C) = \mathrm{Pol}\,\mathrm{Inv}^{(k)} C$ according to Lemma 4.41. □

We shall now present the famous Baker-Pixley theorem which states that a clone is uniquely deretmined by its k-ary invariant relations whenever it contains a special operation. An operation f of arity $\geqslant 3$ is called a *near-unanimity operation* if

$$f(y,x,x,\ldots,x,x) = f(x,y,x,\ldots,x,x) = \ldots = f(x,x,x,\ldots,x,y) = x$$

for all $x,y \in A$.

Theorem 4.43 (Baker, Pixley 1975, [3]) *Let $k \geqslant 2$ and suppose that a clone C contains an $(k+1)$-ary near-unanimity operation. Then $C = \operatorname{Pol}\operatorname{Inv}^{(k)} C$.*

Proof. Let $\nu \in C^{(k+1)}$ be a near-unanimity operation and let us show that $\operatorname{Pol}\operatorname{Inv}^{(k)} C \subseteq \operatorname{Pol}\operatorname{Inv} C$ since the other inclusion always holds.

Take any $f \in \operatorname{Pol}\operatorname{Inv}^{(k)} C$, any positive integer h and any $\rho \in \operatorname{Inv}^{(h)} C$. If $h = k$ we trivially have that f preserves ρ. If $h < k$ it suffices to note that $\rho \times A^{k-h} \in \operatorname{Inv}^{(k)} C$ and that f preserves ρ if and only if f preserves $\rho \times A^{k-h}$. Finally, assume $h > k$. For every sequence of indices $1 \leqslant i_1 < \ldots < i_s \leqslant h$ let $\rho_{i_1\ldots i_s} = \operatorname{pr}_{i_1\ldots i_s}(\rho)$. By the assumption, f preserves every $\rho_{i_1\ldots i_k}$. Let us show that then f has to preserve ρ. We demonstrate the main idea by considering a special case of $k = 2$. Let us start by showing that f preserves ρ_{ijl} for all $i < j < l$. Take any $(u_1,v_1,w_1),\ldots,(u_n,v_n,w_n) \in \rho_{ijl}$ and let $u = f(u_1,\ldots,u_n)$, $v = f(v_1,\ldots,v_n)$, $w = f(w_1,\ldots,w_n)$. Since f preserves ρ_{ij}, ρ_{il} and ρ_{jl} we have that $(u,v) \in \rho_{ij}$, $(u,w) \in \rho_{il}$ and $(v,w) \in \rho_{jl}$. But ρ_{ij}, ρ_{il} and ρ_{jl} are projections of ρ_{ijl}, so there exist x, y, $z \in A$ such that

$$(u,v,x) \in \rho_{ijl}, \quad (u,y,w) \in \rho_{ijl}, \quad (z,v,w) \in \rho_{ijl}.$$

Now, $\nu \in C$ preserves all relations in $\operatorname{Inv} C$ and in particular it preserves ρ_{ijl} so

$$(\nu(u,u,z), \nu(v,y,v), \nu(x,w,w)) = (u,v,w) \in \rho_{ijl}.$$

This shows that f preserves all ρ_{ijl}. Next, let us show that f preserves ρ_{ijlm} for all $i < j < l < m$. Take any $(u_1,v_1,w_1,p_1),\ldots,(u_n,v_n,w_n,p_n) \in \rho_{ijlm}$ and let $u = f(u_1,\ldots,u_n)$, $v = f(v_1,\ldots,v_n)$, $w = f(w_1,\ldots,w_n)$, $p = f(p_1,\ldots,p_n)$. Since f preserves ρ_{ijl}, ρ_{ijm} and ρ_{ilm} we have that $(u,v,w) \in \rho_{ijl}$, $(u,v,p) \in \rho_{ijl}$ and $(u,w,p) \in \rho_{ilm}$. But ρ_{ijl}, ρ_{ijm} and ρ_{ilm} are projections of ρ_{ijlm}, so there exist x, y, $z \in A$ such that

$$(u,v,w,x) \in \rho_{ijlm}, \quad (u,v,y,p) \in \rho_{ijlm}, \quad (u,z,w,p) \in \rho_{ijlm},$$

whence

$$(\nu(u,u,u), \nu(v,v,z), \nu(w,y,w), \nu(x,p,p)) = (u,v,w,p) \in \rho_{ijlm}.$$

Now, using induction on s we can show that f preserves $\rho_{i_1\dots i_s}$ for all $1 \leqslant i_1 < \dots < i_s \leqslant h$, and hence f preserves $\rho = \rho_{12\dots h}$. Therefore, $f \in \operatorname{Pol}\operatorname{Inv} C = C$ and this completes the proof. □

Corollary 4.44 *Let C be a clone that contains a near-unanimity operation. Then the order-filter $\uparrow C = \{D \subseteq \mathscr{O}_A : D \text{ is a clone and } D \supseteq C\}$ is finite.*

Proof. Let $v \in C$ be a near-unanimity operation and let $k = \operatorname{ar}(v)$. Then $v \in D$ for every $D \in \uparrow C$, so $D = \operatorname{Pol}\operatorname{Inv}^{(k)} D$ for every $D \in \uparrow C$ and hence the mapping $\varphi : \uparrow C \to \mathscr{P}(\mathscr{R}_A^{(k)})$ given by $\varphi(D) = \operatorname{Inv}^{(k)} D$ is injective. Since $\mathscr{P}(\mathscr{R}_A^{(k)})$ is finite, $\uparrow C$ is also finite. □

Corollary 4.45 *Every clone containing a near-unanimity operation is finitely generated.*

Proof. Let C be a clone and let $v \in C$ be a near-unanimity operation. Let $N = \operatorname{Cln}(v)$. If $C = N$ we are done. Assume, therefore, that $C \supset N$. According to Corollary 4.44 the order-filter $\uparrow N$ is finite, so the interval $[N, C]$ is also finite, say, $[N, C] = \{N, D_1, \dots, D_k, C\}$. For every $i \in \{1, \dots, k\}$ choose an arbitrary $f_i \in C \setminus D_i$, choose $g \in C \setminus N$ and let $F = \operatorname{Cln}(v, g, f_1, \dots, f_k)$. Clearly, $N \subseteq F \subseteq C$ and $F \notin \{N, D_1, \dots, D_k\}$. Therefore, $F = C$ and thus C is finitely generated. □

4.7 Primitive-positive clones

There is another important Galois connection this time between operations on A. We say that *operations $f \in \mathscr{O}_A^{(n)}$ and $g \in \mathscr{O}_A^{(m)}$ commute* if (see Fig. 4.5):

$$\begin{aligned} f(g(a_{11}, a_{12}, \dots, a_{1m}), \dots, g(a_{n1}, a_{n2}, \dots, a_{nm})) = \\ = g(f(a_{11}, a_{21}, \dots, a_{n1}), \dots, f(a_{1m}, a_{2m}, \dots, a_{nm})). \end{aligned}$$

It is easy to see that the following statements are equivalent:

$$f \text{ commutes with } g \Leftrightarrow f \in \operatorname{Pol}\{g^\bullet\} \Leftrightarrow g \in \operatorname{Pol}\{f^\bullet\} \Leftrightarrow g \text{ commutes with } f.$$

The binary relation "… commutes with …" on $\mathscr{O}_A$ generates a Galois connection where the operators $\overrightarrow{\rho}$ and $\overleftarrow{\rho}$ are the same and usually denoted by $(-)^*$. The closure operator is $F \mapsto F^{**}$ and the Galois closed sets are of the form F^*. They are usually referred to as *bicentralizers*, *bicentrally closed sets* or *primitive-positive clones* (the last name is due to Stanley Burris).

a_{11}	a_{12}	$\dots$	a_{1m}	$\overset{g}{\to}$	b_1
a_{21}	a_{22}	$\dots$	a_{2m}	$\overset{g}{\to}$	b_2
$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$
a_{n1}	a_{n2}	$\dots$	a_{nm}	$\overset{g}{\to}$	b_n
$f\downarrow$	$f\downarrow$	$\dots$	$f\downarrow$		$f\downarrow$
c_1	c_2	$\dots$	c_m	$\overset{g}{\to}$	d

Figure 4.5: Commuting operations

Proposition 4.46 (a) *Let $F \subseteq \mathscr{O}_A$ and let $F^\bullet = \{f^\bullet : f \in F\}$. Then $F^* = \mathrm{Pol}(F^\bullet)$.*

(b) *For every primitive-positive clone C of operations on A there is an algebra $\mathbf{A}$ on A such that $C = \bigcup_{n\geqslant 1} \hom(\mathbf{A}^n, \mathbf{A})$.*

Proof. (a) is just a reformulation of the definitions. As for (b), let C be a primitive-positive clone. Then $C = F^*$ for some $F \subseteq \mathscr{O}_A$. It is easy now to see that $C = \bigcup_{n\geqslant 1} \hom(\mathbf{A}^n, \mathbf{A})$ where $\mathbf{A} = (A, F)$. □

As we have seen from Theorems 4.7 and 4.28, both clones of operations and relational clones are subuniverses of algebras of the same type. In both cases we have a binary "composition", two operations that permute variables, one operation that identifies variables and a constant. So, starting from a set of operations F one can first produce a clone $\mathrm{Cln}(F)$ and then interprete it as a set of relations $\mathrm{Cln}(F)^\bullet$, or one can immediately treat F as a set of relations $F^\bullet$ and then produce the relational clone $\mathrm{Clr}(F^\bullet)$, Fig. 4.6. A question arises: what is the relationship

$$\begin{array}{ccc} F & \xrightarrow{\qquad\qquad (-)^\bullet \qquad\qquad} & F^\bullet \\ {\scriptstyle \mathrm{Cln}}\downarrow & & \downarrow{\scriptstyle \mathrm{Clr}\cap\mathscr{O}_A^\bullet} \\ \mathrm{Cln}(F) & \xrightarrow{(-)^\bullet} \mathrm{Cln}(F)^\bullet \overset{?}{=} & \mathrm{Clr}(F^\bullet)\cap\mathscr{O}_A^\bullet \end{array}$$

Figure 4.6: $\mathrm{Cln}(F)^\bullet$ versus $\mathrm{Clr}(F^\bullet)$

between the two sets of relations? Clearly, $\mathrm{Clr}(F^\bullet)$ contains relations that are not graphs of operations, but what if we compare $\mathrm{Cln}(F)^\bullet$ and $\mathrm{Clr}(F^\bullet)\cap\mathscr{O}_A^\bullet$? The following easy lemma deals with the general case.

Proposition 4.47 *For every $F \subseteq \mathscr{O}_A$ we have $\mathrm{Cln}(F)^\bullet \subseteq \mathrm{Clr}(F^\bullet)\cap\mathscr{O}_A^\bullet$.*

Proof. Having in mind Theorems 4.7 and 4.28, it suffices to note that if $\mathrm{ar}(f) \geqslant 2$

then $(f*g)^\bullet = (g^\bullet)\circ(f^\bullet)$, $\zeta(f)^\bullet = \mathrm{pr}_{n,1,2,\ldots,n-1,n+1}(f^\bullet)$, $\tau(f)^\bullet = \tau(f^\bullet)$, $\triangle(f)^\bullet = \triangle(f^\bullet)$ and $(\pi_1^2)^\bullet = \delta_3^{13|2}$. □

We are now going to show that $\mathrm{Cln}(F)^\bullet = \mathrm{Clr}(F^\bullet)\cap\mathscr{O}_A^\bullet$ if and only if $\mathrm{Cln}(F)$ is a primitive-positive clone.

Lemma 4.48 *Let $F\subseteq\mathscr{O}_A$.*
(a) $(\mathrm{Pol}(F^\bullet))^\bullet = \mathrm{Inv}F\cap\mathscr{O}_A^\bullet$.
(b) $(F^{**})^\bullet = \mathrm{Clr}(F^\bullet)\cap\mathscr{O}_A^\bullet$.
(c) $\mathrm{Clr}(\mathrm{Cln}(F)^\bullet) = \mathrm{Clr}(F^\bullet)$.
(d) If $C=\mathrm{Cln}(F)$ then $F^{**}=C^{**}$.

Proof. *(a)* Take any $g^\bullet\in(\mathrm{Pol}(F^\bullet))^\bullet$. Then $g\in\mathrm{Pol}(F^\bullet)$ if and only if g commutes with every $f\in F$ if and only if every $f\in F$ preserves $g^\bullet$ i.e., $g^\bullet\in\mathrm{Inv}F$.

(b) Since $F^*=\mathrm{Pol}(F^\bullet)$ we have $F^{**}=\mathrm{Pol}(\mathrm{Pol}(F^\bullet)^\bullet)$, so

$$(F^{**})^\bullet = \big(\mathrm{Pol}(\mathrm{Pol}(F^\bullet)^\bullet)\big)^\bullet.$$

Let $C=\mathrm{Pol}(F^\bullet)$. Then according to *(a)*,

$$(F^{**})^\bullet = \big(\mathrm{Pol}(C^\bullet)\big)^\bullet = \mathrm{Inv}C\cap\mathscr{O}_A^\bullet = \mathrm{Inv}\,\mathrm{Pol}(F^\bullet)\cap\mathscr{O}_A^\bullet = \mathrm{Clr}(F^\bullet)\cap\mathscr{O}_A^\bullet.$$

(c) Inclusion $\supseteq$ is obvious and follows from $\mathrm{Cln}(F)\supseteq F$. For the other inclusion, note that $\mathrm{Cln}(F)^\bullet\subseteq\mathrm{Clr}(F^\bullet)$ according to Proposition 4.47, so $\mathrm{Clr}(\mathrm{Cln}(F)^\bullet)\subseteq\mathrm{Clr}(\mathrm{Clr}(F^\bullet))=\mathrm{Clr}(F^\bullet)$.

(d) $(C^{**})^\bullet = \mathrm{Clr}(C^\bullet)\cap\mathscr{O}_A^\bullet = \mathrm{Clr}(\mathrm{Cln}(F)^\bullet)\cap\mathscr{O}_A^\bullet = \mathrm{Clr}(F^\bullet)\cap\mathscr{O}_A^\bullet = (F^{**})^\bullet$. Therefore, $C^{**}=F^{**}$. □

Theorem 4.49 *Let $F\subseteq\mathscr{O}_A$. Then $\mathrm{Cln}(F)$ is a primitive-positive clone if and only if $\mathrm{Cln}(F)^\bullet=\mathrm{Clr}(F^\bullet)\cap\mathscr{O}_A^\bullet$.*

Proof. $(\Rightarrow)$ Let $C=\mathrm{Cln}(F)$ be a primitive-positive clone. Then $C=C^{**}$ and

$$\mathrm{Cln}(F)^\bullet = C^\bullet = (C^{**})^\bullet = \mathrm{Clr}(C^\bullet)\cap\mathscr{O}_A^\bullet = \mathrm{Clr}(\mathrm{Cln}(F)^\bullet)\cap\mathscr{O}_A^\bullet = \mathrm{Clr}(F^\bullet)\cap\mathscr{O}_A^\bullet,$$

by *(b)* and *(c)* of Lemma 4.48.

$(\Leftarrow)$ Let $C=\mathrm{Cln}(F)$. According to Lemma 4.48 and the assumption:

$$(C^{**})^\bullet = \mathrm{Clr}(C^\bullet)\cap\mathscr{O}_A^\bullet = \mathrm{Clr}(\mathrm{Cln}(F)^\bullet)\cap\mathscr{O}_A^\bullet = \mathrm{Clr}(F^\bullet)\cap\mathscr{O}_A^\bullet = \mathrm{Cln}(F)^\bullet = C^\bullet.$$

Therefore, $(C^{**})^\bullet = C^\bullet$, i.e., $C^{**}=C$ and hence C is a primitive-positive clone. □

There is another description of primitive-positive clones that justifies the name. Recall that a *primitive-positive formula over a language* $\mathscr{F} \cup \mathscr{C} \cup \mathscr{Q}$, where $\mathscr{F}$ is a set of function symbols, $\mathscr{C}$ is a set of constant symbols and $\mathscr{Q}$ is a set of relation symbols, is a formula of the form

$$\exists x_1 \exists x_2 \ldots \exists x_m \, (\alpha_1 \wedge \ldots \wedge \alpha_k)$$

where α_i are *atomic formulas*, that is, either formulas of the form $Q(v_1, \ldots, v_n)$ for some $Q \in \mathscr{Q}$ and some variables v_j, or an equality $t(v_1, \ldots, v_n) = s(v_1, \ldots, v_n)$ where t and s are $(\mathscr{F} \cup \mathscr{C})$-terms.

Let F be a set of operations such that (A, F) is an $\mathscr{F}$-algebra. An operation $g \in \mathscr{O}_A^{(n)}$ is *primitive-positive definable over* F (or pp-definable, for short) if there is a primitive-positive formula $\varphi(x_1, \ldots, x_n, y)$ over $\mathscr{F}$ such that

$$g(a_1, \ldots, a_n) = b \quad \text{if and only if} \quad (A, F) \models \varphi[a_1, \ldots, a_n, b]$$

for every $a_1, \ldots, a_n, b \in A$. Then it is easy to show the following statement:

Proposition 4.50 *Let $F \subseteq \mathscr{O}_A$. The set of all operations that are pp-definable over F is a clone of operations on A.*

We are now ready to show that primitive-positive clones are precisely the clones of pp-definable operations.

Theorem 4.51 (Kuznecov) *Let $F \subseteq \mathscr{O}_A$. Then $F^{**} = \{g \in \mathscr{O}_A : g$ is pp-definable over $F\}$.*

Proof. $(\supseteq)$ Suppose $h \in \mathscr{O}_A$ is pp-definable over F. Then there is a primitive positive formula φ over $\mathscr{F}$ that defines h, i.e.

$$h(x_1, \ldots, x_n) = y \quad \text{if and only if} \quad (A, F) \models \varphi(x_1, \ldots, x_n, y).$$

Let $\mathbf{x} = (x_1, \ldots, x_n)$, $\mathbf{A} = (A, F)$ and let $\varphi^{\mathbf{A}}$ denote the interpretation of φ in $\mathbf{A}$. Clearly, $\varphi^{\mathbf{A}} = h^\bullet$. Without loss of generality we may assume that

$$\varphi(\mathbf{x}, y) = (\exists \mathbf{z}) \bigwedge_{i=1}^{s} (f_i(\mathbf{x}, y, \mathbf{z}) = g_i(\mathbf{x}, y, \mathbf{z}))$$

for appropriately chosen $f_i, g_i \in C = \mathrm{Cln}(F)$. Let us show that $\varphi^{\mathbf{A}} \in \mathrm{Clr}(C^\bullet)$. From $f_i, g_i \in C$ it follows that $f_i^\bullet, g_i^\bullet \in C^\bullet$, so $f_i^\bullet \times g_i^\bullet \in \mathrm{Clr}(C^\bullet)$ for all i. For appropriately chosen diagonals δ_i and mappings α_i we have

$$(f_i(\mathbf{x}, y, \mathbf{z}) = g_i(\mathbf{x}, y, \mathbf{z}))^{\mathbf{A}} = \mathrm{pr}_{\alpha_i}(\delta_i \cap (f_i^\bullet \times g_i^\bullet)) \in \mathrm{Clr}(C^\bullet).$$

Next, we have

$$\left(\bigwedge_{i=1}^{s}(f_i(\mathbf{x},y,\mathbf{z})=g_i(\mathbf{x},y,\mathbf{z}))\right)^{\mathbf{A}}=\bigcap_{i=1}^{s}\mathrm{pr}_{\alpha_i}(\delta_i\cap(f_i^\bullet\times g_i^\bullet))\in\mathrm{Clr}(C^\bullet),$$

and thus

$$\varphi^{\mathbf{A}}=\mathrm{pr}_{\xi_1\dots\xi_b}\left(\bigcap_{i=1}^{s}\mathrm{pr}_{\alpha_i}(\delta_i\cap(f_i^\bullet\times g_i^\bullet))\right)\in\mathrm{Clr}(C^\bullet)$$

for appropriately chosen $\xi_1,\dots,\xi_b$. This shows that $h^\bullet=\varphi^{\mathbf{A}}\in\mathrm{Clr}(C^\bullet)$. Therefore, $h\in C^{**}=F^{**}$.

($\subseteq$) Take any $h\in C^{**}$. Then $f^\bullet\in(C^{**})^\bullet=\mathrm{Clr}(C^\bullet)\cap\mathscr{O}_A^\bullet$. The proof now follows from Theorem 4.28 having in mind that operations $\circ$, ζ, τ, $\triangle$ and $\delta_3^{12|3}$ can easily be represented by primitive-positive formulas. □

Therefore, only some very special clones are clones of pp-definable functions. Actually, we shall show that there are only finitely many such clones on a finite set. The situation with relational clones, however, is significantly different: every relational clone is a clone of pp-definable relations.

Let Q be a set of relations such that (A,Q) is a $\mathscr{Q}$-relational structure. A relation $\rho\in\mathscr{R}_A^{(n)}$ is *primitive-positive definable over* Q (or pp-definable, for short) if there is a primitive-positive formula $\varphi(x_1,\dots,x_n)$ over $\mathscr{Q}$ such that

$$(a_1,\dots,a_n)\in\rho\quad\text{if and only if}\quad(A,Q)\models\varphi[a_1,\dots,a_n]$$

for all $a_1,\dots,a_n\in A$.

Theorem 4.52 *Let $Q\subseteq\mathscr{R}_A$. Then* $\mathrm{Clr}(Q)=\{\rho\in\mathscr{R}_A:\rho$ *is pp-definable over* $Q\}$.

Theorem 4.53 (Burris, Willard 1987, [16]) *For any fintite set A there are only finitely many primitive-positive clones.*

Proof. Let (A,F) be an algebra of type $\mathscr{F}$. We say that a relation $\rho\in\mathscr{R}_A^{(4)}$ is definable by a principal congruence formula w.r.t (A,F) if there is a principal congruence formula ψ such that

$$(a,b,c,d)\in\rho\quad\text{if and only if}\quad(A,F)\models\psi[a,b,c,d],$$

for all $a,b,c,d\in A$.

Now let $\mathbf{A}_1=(A,F_1)$ be an algebra of type $\mathscr{F}_1$ and $\mathbf{A}_2=(A,F_2)$ an algebra of type $\mathscr{F}_2$, and assume that the following two conditions hold:

(1) $\rho \in \mathscr{R}_A^{(4)}$ is definable by a principal congruence formula w.r.t $\mathbf{A}_1$ if and only if ρ is definable by a principal congruence formula w.r.t $\mathbf{A}_2$; and

(2) $f \in \bigcup_{i=1}^{|A|} \mathscr{O}_A^{(i)}$ is a term function of algebra $\mathbf{A}_1$ if and only if f is a term function of algebra $\mathbf{A}_2$.

Then we are going to show that $\hom(\mathbf{A}_1^k, \mathbf{A}_1) = \hom(\mathbf{A}_2^k, \mathbf{A}_2)$, for every $k \geqslant 1$.

Let $k \geqslant 1$ be arbitrary and let us first show that (1) implies $\mathrm{Con}(\mathbf{A}_1^k) = \mathrm{Con}(\mathbf{A}_2^k)$. Clearly, it suffices to show that the principal congruences coincide. So, let $\Theta_1(\mathbf{c}, \mathbf{d})$ be a principal congruence of $\mathbf{A}_1^k$ generated by $\mathbf{c}, \mathbf{d} \in A^k$. Then there is a principal congruence formula ψ such that

$$(\mathbf{a}, \mathbf{b}) \in \Theta_1(\mathbf{c}, \mathbf{d}) \quad \text{if and only if} \quad \mathbf{A}_1^k \models \psi[\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}].$$

Since principal congruence formulas respect homomorphisms we have that

$$\mathbf{A}_1 \models \psi[a_1, b_1, c_1, d_1], \ldots, \mathbf{A}_1 \models \psi[a_k, b_k, c_k, d_k],$$

where $\mathbf{a} = (a_1, \ldots, a_k)$, …, $\mathbf{d} = (d_1, \ldots, d_k)$, so the 4-ary relation

$$\rho = \{(a, b, c, d) \in A^4 : \mathbf{A}_1 \models \psi[a, b, c, d]\}$$

contains the quadruples (a_1, b_1, c_1, d_1), …, (a_k, b_k, c_k, d_k). But ρ is obviously definable by a principal congruence formula w.r.t. $\mathbf{A}_1$, so according to (1) ρ is definable by a principal congruence formual w.r.t. $\mathbf{A}_2$. Let $\widehat{\psi}$ be a principle congruence formula which defines ρ w.r.t. $\mathbf{A}_2$. Then

$$\mathbf{A}_2 \models \widehat{\psi}[a_1, b_1, c_1, d_1], \ldots, \mathbf{A}_2 \models \widehat{\psi}[a_k, b_k, c_k, d_k],$$

whence follows that $\mathbf{A}_2^k \models \widehat{\psi}[\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}]$. Therefore, $(\mathbf{a}, \mathbf{b}) \in \Theta_2(\mathbf{c}, \mathbf{d})$. This concludes the proof that $\mathrm{Con}(\mathbf{A}_1^k) = \mathrm{Con}(\mathbf{A}_2^k)$.

Now, take any $\varphi \in \hom(\mathbf{A}_1^k, \mathbf{A}_1)$ and assume that $\varphi \notin \hom(\mathbf{A}_2^k, \mathbf{A}_2)$. From $\varphi \in \hom(\mathbf{A}_1^k, \mathbf{A}_1)$ it follows that $\ker \varphi \in \mathrm{Con}(\mathbf{A}_1^k) = \mathrm{Con}(\mathbf{A}_2^k)$, while $\varphi \notin \hom(\mathbf{A}_2^k, \mathbf{A}_2)$ means that there is a function symbol $f \in \mathscr{F}_2$ and some $u_1, \ldots, u_s \in A^k$ such that

$$\varphi(f^{\mathbf{A}_2^k}(u_1, \ldots, u_s)) \neq f^{\mathbf{A}_2}(\varphi(u_1), \ldots, \varphi(u_s)). \tag{4.2}$$

Let g be a term obtained from f by identifying variables x_i and x_j if and only if $\varphi(u_i) = \varphi(u_j)$. In order to make it easier to follow the proof, we proceed by taking an example. Let $s = 5$ so that f depends on 5 variables $f(x, y, z, v, w)$ and assume that $\varphi(u_2) = \varphi(u_5)$ and $\varphi(u_3) = \varphi(u_4)$. Then

$$g(x, y, z) = f(x, y, z, z, y).$$

Then $f^{\mathbf{A}_2}(\varphi(u_1),\ldots,\varphi(u_s)) = g^{\mathbf{A}_2}(\varphi(u_{i_1}),\ldots,\varphi(u_{i_t}))$ for appropriately chosen indices i_1, ..., i_t. In our example this means that $f^{\mathbf{A}_2}(\varphi(u_1),\varphi(u_2),\varphi(u_3),\varphi(u_4),\varphi(u_5)) = g^{\mathbf{A}_2}(\varphi(u_1),\varphi(u_2),\varphi(u_3))$. Note that $\mathrm{ar}(g) = |\mathrm{im}(\varphi)| \leqslant |A|$.

Next let us show that

$$\varphi(f^{\mathbf{A}_2^k}(u_1,\ldots,u_s)) = \varphi(g^{\mathbf{A}_2^k}(u_{i_1},\ldots,u_{i_t})). \tag{4.3}$$

We again take a look at the example. Since $(u_1,u_1) \in \ker\varphi$, $(u_2,u_2) \in \ker\varphi$, $(u_3,u_3) \in \ker\varphi$, $(u_4,u_3) \in \ker\varphi$, $(u_5,u_2) \in \ker\varphi$ and since $\ker\varphi \in \mathrm{Con}(\mathbf{A}_2^k)$ we have that

$$(f^{\mathbf{A}_2^k}(u_1,u_2,u_3,u_4,u_5), f^{\mathbf{A}_2^k}(u_1,u_2,u_3,u_3,u_2)) \in \ker\varphi$$

i.e.

$$(f^{\mathbf{A}_2^k}(u_1,u_2,u_3,u_4,u_5), g^{\mathbf{A}_2^k}(u_1,u_2,u_3)) \in \ker\varphi,$$

so $\varphi(f^{\mathbf{A}_2^k}(u_1,\ldots,u_s)) = \varphi(g^{\mathbf{A}_2^k}(u_{i_1},\ldots,u_{i_t}))$. Equations (4.2) and (4.3) together with the definition of g imply that

$$\varphi(g^{\mathbf{A}_2^k}(u_{i_1},\ldots,u_{i_t})) \neq g^{\mathbf{A}_2}(\varphi(u_{i_1}),\ldots,\varphi(u_{i_t})). \tag{4.4}$$

Since $g^{\mathbf{A}_2}$ is a term function of $\mathbf{A}_2$ of arity $\leqslant |A|$, from assumption (2) it follows that there is an $\mathscr{F}_1$-term h such that $h^{\mathbf{A}_1} = g^{\mathbf{A}_2}$. From (4.4) if now follows that

$$\varphi(h^{\mathbf{A}_1^k}(u_{i_1},\ldots,u_{i_t})) \neq h^{\mathbf{A}_1}(\varphi(u_{i_1}),\ldots,\varphi(u_{i_t})),$$

which contradicts the fact that $\varphi \in \hom(\mathbf{A}_1^k,\mathbf{A}_1)$ and h is an $\mathscr{F}_1$-term. This completes the proof that assumptions (1) and (2) imply $\hom(\mathbf{A}_1^k,\mathbf{A}_1) = \hom(\mathbf{A}_2^k,\mathbf{A}_2)$.

For a primitive-positive clone C let γ_C be the set of all 4-ary relations on A definable by principal congruence formulas w.r.t. (A,C) and let τ_C be the set of all term functions of (A,C) of arity $\leqslant |A|$, and define the map Φ by $\Phi(C) = (\gamma_C,\tau_C)$. Since $C = \bigcup_{n\geqslant 1}\hom((A,C)^n,(A,C))$, the above discussion actually shows that Φ is an injective mapping. Since there are only finitely many possibilities to chose γ_C and τ_C, it follows that there are only finitely many primitive-positive clones on A. □

4.8 Abstract clones

Each clone C can be understood as a structure with countably many layers $C^{(1)}$, $C^{(2)}$, ..., $C^{(n)}$, ..., substitution operations $S_k^n : C^{(n)} \times (C^{(k)})^n \to C^{(k)}$ and distinguished elements $\pi_1^n,\ldots,\pi_n^n$ of each layer $C^{(n)}$. An abstrast setting to express this point of view is that of multisorted algebras.

Let S be a nonempty set called the *set of sorts*. An *S-set* (or a *multisorted set*) is any family $(A_s)_{s\in S}$. A multisorted set $A=(A_s)_{s\in S}$ is a *subset of* a multisorted set $B=(B_s)_{s\in S}$, in symbols $A\subseteq B$, if $A_s\subseteq B_s$ for all $s\in S$. An *S-function* $f:A\to B$ between S-sets $A=(A_s)_{s\in S}$ and $B=(B_s)_{s\in S}$ is any family of maps $(f_s)_{s\in S}$ such that $f_s:A_s\to B_s$ is a (usual) mapping for all $s\in S$. An S-function $f:A\to B$ is *bijective* if all f_s's are bijective. An *S-equivalence relation* $\theta=(\theta_s)_{s\in S}$ on an S-set $A=(A_s)_{s\in S}$ is an S-set $(\theta_s)_{s\in S}$ such that θ_s is an equivalence relation on A_s for all $s\in S$. By A/θ we denote the S-set $(A_s/\theta_s)_{s\in S}$.

For a string of sorts $w=(s_1,\ldots,s_n)\in S^n$ and a sort $s_0\in S$, a *(w,s_0)-operation* (or a *multisorted operation*) on $(A_s)_{s\in S}$ is any mapping $f:A_{s_1}\times\ldots\times A_{s_n}\to A_{s_0}$. In that case w is said to be the *arity* of f. If $w=()$ is the empty string of sorts, then a $((),s_0)$-operation is just an element of A_{s_0}. Therefore, $((),s_0)$-operations correspond to constants of sort s_0.

An *S-signature* is a set Σ of pairs (w,s) where w is a string of sorts from S and $s\in S$. For a signature Σ, a *Σ-multisorted algebra* is a pair (A,F) where $A=(A_s)_{s\in S}$ is an S-set and F is a set of multisorted operations on A such that for every $\sigma=(w,s)\in\Sigma$ there is exactly one (w,s)-operation $f_\sigma\in F$ and there are no other multisorted operations in F.

Let (A,F^A) and (B,F^B) be Σ-multisorted algebras and let $h:A\to B$ be an S-function. Then *h is a Σ-homomorphism* if

$$h_{s_0}(f^A_\sigma(a_1,\ldots,a_n))=f^B_\sigma(h_{s_1}(a_1),\ldots,h_{s_n}(a_n))$$

for every $\sigma=((s_1,\ldots,s_n),s_0)\in\Sigma$ and all $a_i\in A_{s_i}$, $1\leqslant i\leqslant n$. A *Σ-isomorphism* is a bijective Σ-homomorphism. We say that (B,F^B) is a *Σ-subalgebra* of (A,F^A) if $B\subseteq A$ and

$$f^B_\sigma(b_1,\ldots,b_n)=f^A_\sigma(b_1,\ldots,b_n)$$

for every $\sigma=((s_1,\ldots,s_n),s_0)\in\Sigma$ and all $b_i\in B_{s_i}$, $1\leqslant i\leqslant n$. An S-equivalence relation θ on A is a *Σ-congruence* of (A,F^A) if for all $\sigma=((s_1,\ldots,s_n),s_0)\in\Sigma$,

$$(a_1,b_1)\in\theta_{s_1},\ldots,(a_n,b_n)\in\theta_{s_n}$$

implies

$$(f^A_\sigma(a_1,\ldots,a_n),f^A_\sigma(b_1,\ldots,b_n))\in\theta_{s_0}.$$

If θ is a Σ-congruence of (A,F^A) then the S-set A/θ is the carrier of a Σ-algebra whose operations are defined by

$$f^{A/\theta}_\sigma(a_1/\theta_{s_1},\ldots,a_n/\theta_{s_n})=f^A_\sigma(a_1,\ldots,a_n)/\theta_{s_0}$$

where $\sigma=((s_1,\ldots,s_n),s_0)\in\Sigma$. This algebra is referred to as the *factor algebra* and denoted also by A/θ. One can now show that all the facts from universal algebra easily carry over to multisorted algebras.

Definition 4.54 An *abstract clone* is a multisorted algebra whose carrier is an $\mathbb{N}$-set $(A_n)_{n\in\mathbb{N}}$, with operations S^n_k of signature $((n,\underbrace{k,\ldots,k}_{n}),k)$, and with constants $e^n_i \in A_n$, $1 \leqslant i \leqslant n$, which satisfies the following identities for all reasonable choices of i, k, m and n:

(AC1) $S^n_n(f,e^n_1,\ldots,e^n_n) = f$,

(AC2) $S^n_k(e^n_i,f_1,\ldots,f_n) = f_i$, and

(AC3) $S^k_m(S^n_k(f,g_1,\ldots,g_n),h_1,\ldots,h_k) =$
$= S^n_m(f,S^k_m(g_1,h_1,\ldots,h_k),\ldots,S^k_m(g_n,h_1,\ldots,h_k))$.

Example 4.55 (a) Clearly, every clone of operations C is an abstract clone. The carrier of the algebra is $(C^{(n)})_{n\in\mathbb{N}}$, the operations S^n_k are given by $S^n_k(f,g_1,\ldots,g_n) = f(g_1,\ldots,g_n)$ and the constants are $e^n_i = \pi^n_i$.

(b) There are abstract clones that are not clones of operations. We start with a straighforward example. Let $\mathscr{F}$ be an algebraic type, let $X_n = \{x_1,\ldots,x_n\}$, $n \in \mathbb{N}$, be an increasing chain of finite sets of variables, let $X = \bigcup_{n\geqslant 1} X_n$ and let $T_{\mathscr{F}}(X_n)$ denote the absolutely free $\mathscr{F}$-algebra over the set of variables X_n. Then $\mathrm{Term}_{\mathscr{F}}(X) = (T_{\mathscr{F}}(X_n))_{n\in\mathbb{N}}$ is the carrier of an abstract clone whose constants are given by $e^n_i = x_i$ and the superposition operations are given by substituting terms for variables. Note that if $\mathbf{A} = (A,F)$ is an $\mathscr{F}$-algebra and $\iota : \mathscr{F} \to F$ is the interpretation of operation symbols, then ι extends to a clone homomorphism $\iota^{\#} : \mathrm{Term}_{\mathscr{F}}(X) \to \mathrm{Cln}(\mathbf{A})$, where $\mathrm{Cln}(\mathbf{A})$ denotes the clone of term-operations of the algebra $\mathbf{A}$.

(c) Finally, there exist abstract clones which are not just clones of operations in a fancy robe. Let X be a nonempty set and let A_n, $n \in \mathbb{N}$, be the set of all mappings $f : X \to \{1,\ldots,n\} \times X$. Let $e^n_i \in A_n$ be the following mapping: $e^n_i(x) = (i,x)$. For $g_1, \ldots, g_n \in A_k$ define $[g_1,\ldots,g_n] : \{1,\ldots,n\} \times X \to \{1,\ldots,k\} \times X$ by $[g_1,\ldots,g_n](i,x) = g_i(x)$ and let $S^n_k(f,g_1,\ldots,g_n) = [g_1,\ldots,g_n] \circ f$. Then $\Big((A_n)_{n\in\mathbb{N}}, (S^n_k)_{n,k\in\mathbb{N}}, (e^n_i)_{\substack{n,i\in\mathbb{N}\\ 1\leqslant i\leqslant n}}\Big)$ is an abstract clone.

Theorem 4.56 *Every abstract clone is isomorphic to a clone of operations (not necessarily on a finite set).*

Proof. Let $(A_n)_{n\in\mathbb{N}}$ be an abstract clone. In order to make it easier to follow the proof we assume that all A_n's are pairwise disjoint and instead of $S^n_k(f,g_1,\ldots,g_n)$ we shall simply write $f(g_1,\ldots,g_n)$. For $a_k \in A_k$ let

$$\begin{aligned} L(a_k) &= \{a_k(e^n_1,\ldots,e^n_k) : n \geqslant k\} \\ &= \{a_k, a_k(e^{k+1}_1,\ldots,e^{k+1}_k), a_k(e^{k+2}_1,\ldots,e^{k+2}_k),\ldots\}. \end{aligned}$$

We say that $L(a_k)$ is a chain, or the chain that starts with a_k. Let $L_n(a_k) = L(a_k) \cap A_n$. Clearly, $L_n(a_k) = \varnothing$ if $n < k$ and $L_n(a_k) = \{a_k(e_1^n, \ldots, e_k^n)\}$ for $n \geqslant k$.

Let us now show that if L and L' are chains such that $L_n = L'_n$ for some $n \in \mathbb{N}$ then $L \subseteq L'$ or $L' \subseteq L$. Assume that $L_n = L'_n = \{a_n\}$, let i be the least integer such that $L_i \neq \varnothing$ and let j be the least integer such that $L'_j \neq \varnothing$ so that $L = L(a_i)$ where $L_i = \{a_i\}$, and $L' = L(a'_j)$ where $L'_j = \{a'_j\}$. Clearly, $i \leqslant n$ and $j \leqslant n$ and without loss of generality we may assume that $i \leqslant j$. Let us show that $L' \subseteq L$.

First, we show that $L_j = L'_j$. Let $L_j = \{a_j\}$. Since $L'_n = \{a_n\}$ and $L' = L(a'_j)$ we have $a_n = a'_j(e_1^n, \ldots, e_j^n)$ and similarly from $L = L(a_i)$ we have $a_n = a_i(e_1^n, \ldots, e_i^n)$. Since $a_j = a_i(e_1^j, \ldots, e_i^j)$ it is easy to see that $a_n = a_j(e_1^n, \ldots, e_j^n)$:

$$\begin{aligned} a_j(e_1^n, \ldots, e_j^n) &= a_i(e_1^j, \ldots, e_i^j)(e_1^n, \ldots, e_j^n) \qquad \text{[by the def. of } a_j\text{]} \\ &= a_i(e_1^j(e_1^n, \ldots, e_j^n), \ldots, e_i^j(e_1^n, \ldots, e_j^n)) \qquad \text{[by (AC3)]} \\ &= a_i(e_1^n, \ldots, e_i^n) \qquad \text{[by (AC2)]} \\ &= a_n. \end{aligned}$$

Therefore, $a_n = a'_j(e_1^n, \ldots, e_j^n) = a_j(e_1^n, \ldots, e_j^n)$, whence

$$a'_j(e_1^n, \ldots, e_j^n)(\underbrace{e_1^j, \ldots, e_j^j, e_j^j, \ldots e_j^j}_{n}) = a_j(e_1^n, \ldots, e_j^n)(\underbrace{e_1^j, \ldots, e_j^j, e_j^j, \ldots e_j^j}_{n})$$

so by (AC3) we have $a'_j(e_1^j, \ldots, e_j^j) = a_j(e_1^j, \ldots, e_j^j)$ and by $(AC1)$ we conclude $a'_j = a_j$, i.e., $L'_j = L_j$. Finally, we show that for all $k \geqslant j$ we have $L_k = L'_k$. Let $L_k = \{a_k\}$ and $L'_k = \{a'_k\}$. Then $a'_k = a'_j(e_1^k, \ldots, e_j^k)$ and $a_k = a_i(e_1^k, \ldots, e_i^k)$. As in the previous paragraph we can show that $a_k = a_j(e_1^k, \ldots, e_j^k)$ so from $a_j = a'_j$ it follows that $a_k = a'_k$. Since $L'_k = \varnothing$ for $k < j$ and $L'_k = L_k$ for $k \geqslant j$ we conclude $L' \subseteq L$.

This shows that every $a \in \bigcup_{n\in\mathbb{N}} A_n$ is contained in finitely many chains $L^1, \ldots, L^s$ and that all these chains are linearly ordered by inclusion, i.e., $L^1 \subseteq \ldots \subseteq L^s$. Therefore, for every $a \in \bigcup_{n\in\mathbb{N}} A_n$ there is a maximal chain that contains it, and we shall denote it by $\overline{a}$. Let

$$\overline{A} = \{\overline{a} : a \in \bigcup_{n\in\mathbb{N}} A_n\}$$

be the set of all maximal chains. For $f \in A_n$ define the n-ary operation $\Omega_f : (\overline{A})^n \to \overline{A}$ on $\overline{A}$ as follows. Take any $\overline{a^1}, \ldots, \overline{a^n} \in \overline{A}$ and find the least $t \geqslant 1$ such that $\overline{a^i} \cap A_t \neq \varnothing$ for all i. Let $\overline{a^i} \cap A_t = \{a_t^i\}$, $i \in \{1, \ldots, n\}$, and set

$$\Omega_f(\overline{a^1}, \ldots, \overline{a^n}) = \overline{f(a_t^1, \ldots, a_t^n)}.$$

Note that for $s \geqslant t$ we have

$$\overline{f(a_t^1,\ldots,a_t^n)} = \overline{f(a_s^1,\ldots,a_s^n)} \tag{4.5}$$

where $\overline{a^i} \cap A_s = \{a_s^i\}$, $i \in \{1,\ldots,n\}$. This follows easily since

$$f(a_s^1,\ldots,a_s^n) = f(a_t^1(e_1^s,\ldots,e_t^s),\ldots,a_t^n(e_1^s,\ldots,e_t^s)) = f(a_t^1,\ldots,a_t^n)(e_1^s,\ldots,e_t^s)$$

and thus $\overline{f(a_t^1,\ldots,a_t^n)} \cap A_s = \{f(a_s^1,\ldots,a_s^n)\}$. Therefore, $f(a_s^1,\ldots,a_s^n)$ and $f(a_t^1,\ldots,a_t^n)$ belong to the same chain and hance have the same maximal chain. To complete the proof, let

$$C = \big\{\Omega_f : f \in \bigcup_{n\in\mathbb{N}} A_n\big\}.$$

At the same time we show that C is a clone and that $\Omega : \bigcup_{n\in\mathbb{N}} A_n \to C : f \mapsto \Omega_f$ is a clone homomorphism. We start with projections:

$$\Omega_{e_i^n}(\overline{a^1},\ldots,\overline{a^n}) = \overline{e_i^n(a_t^1,\ldots,a_t^n)} = \overline{a_t^i} = \overline{a^i},$$

where $t \geqslant 1$ is the least integer such that $\overline{a^i} \cap A_t \neq \varnothing$ for all i, and $\overline{a^i} \cap A_t = \{a_t^i\}$, $i \in \{1,\ldots,n\}$. Therefore, $\Omega_{e_i^n} = \pi_i^n \in C$. Next, take any Ω_f, Ω_{g_1}, …, $\Omega_{g_k} \in C$ where $f \in A_k$ and g_1, …, $g_k \in A_n$. Then

$$\begin{aligned}\Omega_f(\Omega_{g_1},\ldots,\Omega_{g_k})(\overline{a^1},\ldots,\overline{a^n}) &= \Omega_f(\Omega_{g_1}(\overline{a^1},\ldots,\overline{a^n}),\ldots,\Omega_{g_k}(\overline{a^1},\ldots,\overline{a^n})) \\ &= \Omega_f(\overline{g_1(a_t^1,\ldots,a_t^n)},\ldots,\overline{g_k(a_t^1,\ldots,a_t^n)}),\end{aligned}$$

where $t \geqslant 1$ is the least integer such that $\overline{a^i} \cap A_t \neq \varnothing$ for all i, and $\overline{a^i} \cap A_t = \{a_t^i\}$, $i \in \{1,\ldots,n\}$. From (4.5) and (AC1)–(AC3) it now follows that

$$\begin{aligned}\Omega_f(\,\overline{g_1(a_t^1,\ldots,a_t^n)},\ldots,\overline{g_k(a_t^1,\ldots,a_t^n)}\,) &= \overline{f(g_1(a_t^1,\ldots,a_t^n),\ldots,g_k(a_t^1,\ldots,a_t^n))} \\ &= \overline{f(g_1,\ldots,g_k)(a_t^1,\ldots,a_t^n)} \\ &= \Omega_{f(g_1,\ldots,g_k)}(\overline{a^1},\ldots,\overline{a^n}).\end{aligned}$$

Therefore, $\Omega_f(\Omega_{g_1},\ldots,\Omega_{g_k}) = \Omega_{f(g_1,\ldots,g_k)} \in C$. So, C is a clone and Ω is a clone homomorphism. Clearly, Ω is onto and in order to show that Ω is an isomorphism we still have to show that Ω is injective. Let $f,g \in A_n$ and let $\Omega_f = \Omega_g$. Then

$$\overline{f} = \overline{f(e_1^n,\ldots e_n^n)} = \Omega_f(\overline{e_1^n},\ldots,\overline{e_n^n}) = \Omega_g(\overline{e_1^n},\ldots,\overline{e_n^n}) = \overline{g(e_1^n,\ldots e_n^n)} = \overline{g}.$$

So, $\overline{f} = \overline{g}$ and from $f,g \in A_n$ and $|\overline{f} \cap A_n| = 1 = |\overline{g} \cap A_n|$ it follows $f = g$. This completes the proof. □

Recall that each algebraic type $\mathscr{F}$ corresponds to an abstract clone whose elements are terms: let $X_n = \{x_1, \ldots, x_n\}$, $n \in \mathbb{N}$, be an increasing chain of finite sets of variables, let $T_{\mathscr{F}}(X_n)$ denote the absolutely free $\mathscr{F}$-algebra over the set of variables X_n and let $X = \bigcup_{n\in\mathbb{N}} X_n$; then $\mathrm{Term}_{\mathscr{F}}(X) = (T_{\mathscr{F}}(X_n))_{n\in\mathbb{N}}$ is the carrier of an abstract clone whose constants are $e_i^n = x_i$ and the superposition operations are given by substituting terms for variables.

Let $\mathscr{V}$ be a variety of type $\mathscr{F}$ and let $\mathrm{Eq}(\mathscr{V})$ denote the equational theory of $\mathscr{V}$, that is

$$\mathrm{Eq}(\mathscr{V}) = \bigcup_{n\in\mathbb{N}} \mathrm{Eq}_n(\mathscr{V})$$

where

$$\mathrm{Eq}_n(\mathscr{V}) = \{(p,q) : p,q \in T_{\mathscr{F}}(X_n) \text{ and } \mathscr{V} \models p \approx q\}.$$

Then $\mathrm{Eq}(\mathscr{V})$ is a congruence of $\mathrm{Term}_{\mathscr{F}}(X)$ and the factor-clone $\mathrm{Term}_{\mathscr{F}}(X)/\mathrm{Eq}(\mathscr{V})$ is just another representation of the free $\mathscr{V}$-algebra on a countable set of generators. In particular,

Lemma 4.57 *If $\mathbf{A} = (A, F)$ is an $\mathscr{F}$-algebra and $\mathscr{V}(\mathbf{A})$ the variety generated by $\mathbf{A}$ then* $\mathrm{Term}_{\mathscr{F}}(X)/\mathrm{Eq}(\mathscr{V}(\mathbf{A})) \cong \mathrm{Cln}(\mathbf{A})$.

Proof. This is straightforward, and we include the proof just to demonstrate the language of abstract clone theory. Recall that if $\iota : \mathscr{F} \to F$ is the interpretation of fundamental operation symbols that gives rise to $\mathbf{A}$, then ι extends to a clone homomorphism $\iota^{\#} : \mathrm{Term}_{\mathscr{F}}(X) \to \mathrm{Cln}(\mathbf{A})$, and this homomorphism is onto. So, by the First Isomorphism Theorem it suffices to show that $\ker(\iota^{\#}) = \mathrm{Eq}(\mathscr{V}(\mathbf{A}))$, and this is easy: $\iota^{\#}(p) = \iota^{\#}(q)$ if and only if $p^{\mathbf{A}} = q^{\mathbf{A}}$ if and only if $\mathbf{A} \models p \approx q$ if and only if $\mathscr{V}(\mathbf{A}) \models p \approx q$ if and only if $(p,q) \in \mathrm{Eq}(\mathscr{V}(\mathbf{A}))$. □

In [31] A. Knoebel considered maximal clones C on a finite set A as algebras (A, C) and located these algebras in the lattice of varieties of the appropriate similarity type. It turns out that any such algebra (A, C) generates a variety whose subvarieties form a chain of length 1, 2, 4 or 5 under inclusion.

We say that an $\mathscr{F}$-variety is *minimal* if it is an atom in the lattice of all $\mathscr{F}$-varieties. It is easy to see that the congruence lattice of $\mathrm{Term}_{\mathscr{F}}(X)$ is dually isomorphic to the lattice of $\mathscr{F}$-varieties and hence a variety $\mathscr{V}$ is minimal if and only if $\mathrm{Term}_{\mathscr{F}}(X)/\mathrm{Eq}(\mathscr{V})$ has no nontrivial congruences.

Theorem 4.58 (A. Knoebel 1985 [31]) *Let $\leqslant$ be a bounded partial order on a finite set A and let $\mathbf{A}_{\leqslant} = (A, \mathrm{Pol}\{\leqslant\})$. Then the variety $\mathscr{V}(\mathbf{A}_{\leqslant})$ is minimal.*

Proof. Let $\mathscr{F}$ be an algebraic type chosen so that $\mathbf{A}_{\leqslant}$ is an $\mathscr{F}$-algebra. In order to show that $\mathscr{V}(\mathbf{A}_{\leqslant})$ is a minimal $\mathscr{F}$-variety, we show that $\mathrm{Term}_{\mathscr{F}}(X)/\mathrm{Eq}(\mathscr{V}(\mathbf{A}_{\leqslant}))$

has no nontrivial congruences. Since $\mathrm{Term}_{\mathscr{F}}(X)/\mathrm{Eq}(\mathscr{V}(\mathbf{A}_{\leqslant})) \cong \mathrm{Cln}(\mathbf{A}_{\leqslant})$ (Lemma 4.57) and $\mathrm{Cln}(\mathbf{A}) = \mathrm{Pol}\{\leqslant\}$, we are done if we manage to show that $\mathrm{Pol}\{\leqslant\}$ has no nontrivial congruences.

Let $\Theta^0 = (\Theta^0_n)_{n\in\mathbb{N}}$ and $\Theta^1 = (\Theta^1_n)_{n\in\mathbb{N}}$ denote, respectively, the least and the greatest congruence on $\mathscr{O}_A$:

$$\Theta^0_n = \{(f,f) : f \in \mathscr{O}^{(n)}_A\} \quad \text{and} \quad \Theta^1_n = \{(f,g) : f,g \in \mathscr{O}^{(n)}_A\}$$

and let $\theta = (\theta_n)_{n\in\mathbb{N}}$ be a congruence on $\mathrm{Pol}\{\leqslant\}$ distinct from Θ^0. Then there is an n such that $\theta_n \neq \Theta^0_n$, i.e. there exist $n \in \mathbb{N}$ and $f,g \in \mathrm{Pol}^{(n)}\{\leqslant\}$ such that $f \neq g$ and $(f,g) \in \theta_n$. Let us first show that this implies that for every k we have $\theta_k \neq \Theta^0_k$.

Take $a_1,\dots,a_n \in A$ so that $p = f(a_1,\dots,a_n) \neq g(a_1,\dots,a_n) = q$. Let c_a denote the constant unary mapping $x \mapsto a$. Clearly, $c_a \in \mathrm{Pol}\{\leqslant\}$ for all $a \in A$. Then from $f\ \theta_n\ g$, $c_{a_1}\ \theta_1\ c_{a_1}$, …, $c_{a_n}\ \theta_1\ c_{a_n}$ and the fact that θ is a clone congruence it follows that

$$c_p = f(c_{a_1},\dots,c_{a_n})\ \theta_1\ g(c_{a_1},\dots,c_{a_n}) = c_q.$$

Since $p \neq q$, we have $p \not\leqslant q$ or $q \not\leqslant p$, so assume that $q \not\leqslant p$. Then there is an $h \in \mathrm{Pol}^{(1)}\{\leqslant\}$ such that $h(p) = 0$ and $h(q) = 1$, where 0 and 1 denote the least and the greatest element of $\leqslant$. Now, $h\ \theta_1\ h$ and $c_p\ \theta_1\ c_q$, so $h(c_p)\ \theta_1\ h(c_q)$, i.e. $c_0\ \theta_1\ c_1$. But then for every $k \in \mathbb{N}$ we have that $c_0(\pi^k_1) \neq c_1(\pi^k_1)$ and $c_0(\pi^k_1)\ \theta_k\ c_1(\pi^k_1)$. Therefore, $\theta_k \neq \Theta^0_k$ for all $k \in \mathbb{N}$.

Finally, let us show that for all $k \in \mathbb{N}$, if $\theta_k \neq \Theta^0_k$ then $\theta_k = \Theta^1_k$. Take any $s \in \mathrm{Pol}^{(k)}\{\leqslant\}$ and consider $t \in \mathscr{O}^{(k+1)}_A$ defined by

$$t(x_1,\dots,x_k,x_{k+1}) = \begin{cases} 0, & x_{k+1} \neq 1 \\ s(x_1,\dots,x_k), & x_{k+1} = 1. \end{cases}$$

Then it is easy to see that $t \in \mathrm{Pol}^{(k+1)}\{\leqslant\}$ and that

$$s = t(\pi^k_1,\dots,\pi^k_k,c_1(\pi^k_1))\ \theta_k\ t(\pi^k_1,\dots,\pi^k_k,c_0(\pi^k_1)) = c_0(\pi^k_1).$$

Therefore, $s\ \theta_k\ c_0(\pi^k_1)$ for every $s \in \mathrm{Pol}^{(k)}\{\leqslant\}$, whence $\theta_k = \Theta^1_k$. □

Chapter 5

Minimal clones and CSP

One of the main concerns in theoretical computer science is to understand which computational problems are tractable, and which problems are hard to solve. Here, "tractable" means that instances of the problem can be solved within a reasonable amount of computational resources and time. In this text we designate problems as tractable if there exists a polynomial time algorithm, whereas hard are those that are NP-hard.

The constraint satisfaction problem was introduced by Montanari in 1974 and has been widely studied [42]. Several frameworks to formalize the notion of constraint satisfaction have been proposed, most prominantly the class CSP of constraint satisfaction problems that are defined as homomorphism problems. Such problems are defined by a relational structure, the so-called template of the constraint satisfaction problem. Constraint satisfaction problems are computational problems that occur in many areas of computer science, graph theory, boolean satisfiability and database theory.

One fundamental open research problem in this area is to characterise exactly the forms of constraint relations which give rise to tractable problem classes. This problem is important from a theoretical perspective, as it helps to clarify the boundary between tractability and intractability in a wide range of combinatorial search problems.

The problem of characterising the tractable cases was completely solved for the important special case of Boolean constraint satisfaction problems by Schaefer in 1978 [55]. Schaefer established that for Boolean constraint satisfaction problems (which he called Generalised Satisfiability Problems) there are exactly six different families of tractable constraints, and any problem involving constraints not contained in one of these six families is NP-complete. This important result is known as Schaefer's Dichotomy Theorem. In 2002. Bulatov managed to ob-

tain a complete classification for the complexity of constraints on a three-element set [11]. There is still no complete classiffcation for the complexity of constraints over finite sets with more than three elements, and no dichotomy has so far been established for arbitrary finite sets, although it has been conjectured that in case of an arbitrary finite templates the class of CSP problems satisfies the dichotomy principle.

5.1 Introduction

Let X be a finite set of variables, and A a finite set of values. A mapping $f : X \to A$ will be referred to as a *valuation*. For $h \geqslant 1$, an *h-ary constraint* is an ordered pair $(\mathbf{x}, \rho)$ where $\mathbf{x} \in X^h$ and $\rho \in \mathscr{R}_A^{(h)}$. We say that a valuation $f : X \to A$ satisfies a constraint $(\mathbf{x}, \rho)$ if $f(\mathbf{x}) \in \rho$.

The *constraint satisfaction problem (CSP)* is a class of decision problems $(X, A, \mathscr{C})$ where X and A are finite sets, $\mathscr{C} = \{(\mathbf{x}^1, \rho_1), \ldots, (\mathbf{x}^k, \rho_k)\}$ is a finite class of constraints over the set of variables X and the set of values A, and the problem is to decide whether there exists a valuation $f : X \to A$ which satisfies each constraint in $\mathscr{C}$, i.e., such that $f(\mathbf{x}^i) \in \rho_i$, for all i?

Since both X and A are always finite, it is clear that every instance of CSP is decidable. The real problem is, therefore, to establish the complexity of each particular decision problem.

Example 5.1 An instance of a k-SATISFIABILITY problem asks whether a propositional formula

$$F(x_1, \ldots, x_n) = \bigwedge_{i=1}^{m} (x_{i1}^{\varepsilon_{i1}} \vee \ldots \vee x_{ik}^{\varepsilon_{ik}})$$

in its conjuctive normal form over a set of variables $\{x_1, \ldots, x_n\}$ is satisfiable, i.e. whether there exists an assignment of truth values to variables that makes the formula true. Here, $\varepsilon_{ij} \in \{0, 1\}$ and we follow the convention that

$$x^{\varepsilon} = \begin{cases} x, & \varepsilon = 1 \\ \neg x, & \varepsilon = 0. \end{cases}$$

The CSP interpretation of the problem is straightforward. Let $X = \{x_1, \ldots, x_n\}$ and let $A = \{0, 1\}$. For each conjunct $x_{i1}^{\varepsilon_{i1}} \vee \ldots \vee x_{ik}^{\varepsilon_{ik}}$ take a constraint $S_i = ((x_{i1}, \ldots, x_{ik}), \rho_i)$ where

$$\rho_i = A^k \setminus \{(1 - \varepsilon_{i1}, \ldots, 1 - \varepsilon_{ik})\}$$

Note that ρ_i is the set of all the k-tuples $(a_1, \ldots, a_k) \in A^k$ such that the conjunct $x_{i1}^{\varepsilon_{i1}} \vee \ldots \vee x_{ik}^{\varepsilon_{ik}}$ evaluates to 1 under the assignment of truth values $x_{i1} = a_1$, ...,

$x_{ik} = a_k$. Now put $\mathscr{C}_F = \{S_1, \ldots, S_m\}$. Then this particular instance of CSP has a solution if and only if the formula F is satisfiable.

Example 5.2 An instance of a k-GRAPHCOLORABILITY problem asks whether a given graph G is k-colourable. The CSP interpretation of the problem is again straightforward. Let $X = \{x_1, \ldots, x_n\}$, be the set of vertices of G, let $\{e_1, \ldots, e_m\}$ be the set of edges of G where $e_i = \{u_i, v_i\} \subseteq X$ with $u_i \neq v_i$. Let $A = \{1, 2, \ldots, k\}$ and $\nu_A = \{(x,y) \in A^2 : x \neq y\}$. For each edge $e_i = \{u_i, v_i\}$ we take a constraint

$$S_i = ((u_i, v_i), \nu_A)$$

and put $\mathscr{C}_G = \{S_1, \ldots, S_m\}$. Clearly, this instance of CSP has a solution if and only if G is k-colourable.

Example 5.3 An instance of a k-CLIQUE problem asks whether a given graph $G = (V,E)$ contains a k-clique. The CSP interpretation of the problem is slightly more involved. Let $X = \{x_1, \ldots, x_k\}$, let $A = V = \{a_1, \ldots, a_n\}$ be the set of vertices of G, and let

$$\varepsilon_G = \bigcup_{\{u,v\} \in E} \{(u,v), (v,u)\}.$$

Now, for a pair of distinct indices $i, j \in \{1, 2, \ldots, k\}$, let

$$S_{ij} = \{((x_i, x_j), \nu_A), ((x_i, x_j), \varepsilon_G)\},$$

and put

$$\mathscr{C}_G = \bigcup \{S_{ij} : i, j \in \{1, 2, \ldots, k\}, i \neq j\}.$$

Clearly, if G has a k-clique spanned by $b_1, \ldots, b_k \in V$ then $f = \left(\begin{smallmatrix} x_1 & x_2 & \ldots & x_k \\ b_1 & b_2 & \ldots & b_k \end{smallmatrix}\right)$ is a valuation that satisfies every constraint in $\mathscr{C}_G$. Conversely, if a valuation $f : X \to A$ satisfies every constraint in $\mathscr{C}_G$, then f is injective due to the set of constraints

$$\mathscr{C}'_G = \{((x_i, x_j), \nu_A) : i, j \in \{1, 2, \ldots, k\}, i \neq j\}$$

and $f(X)$ spans a complete subgraph of G due to the set of constraints

$$\mathscr{C}''_G = \{((x_i, x_j), \varepsilon_G) : i, j \in \{1, 2, \ldots, k\}, i \neq j\}.$$

Therefore, $f(X)$ is the set of vertices of a k-clique in G.

Example 5.4 An instance of a HAMILTONIAN problem asks whether a given graph $G = (V,E)$ contains a Hamiltonian cycle. For a CSP interpretation of the problem let $X = \{x_1,\ldots,x_n\}$, where n is the number of vertices of G, and let $A = V = \{a_1,\ldots,a_n\}$ be the set of vertices of G. Define ε_G and S_{ij} as in Example 5.3 and let

$$\mathscr{C}_G = S_{12} \cup S_{23} \cup \ldots \cup S_{n-1,n} \cup S_{n1}.$$

Now, if a valuation $f : X \to A$ satisfies every constraint in $\mathscr{C}_G$, then f is injective, $|f(X)| = n$ and thus $f(X)$ spans a cycle in G which contains every vertex of G. Therefore, $f(X)$ is the set of vertices of a Hamiltonian cycle in G.

It is well known that 3-SATISFIABILITY is NP-complete. It follows from Example 5.1 then that the general CSP is also NP-complete. However, certain restrictions may affect the complexity of CSP. One of the possible natural ways to restrict CSP is to limit the scope of relations which can appear as constraints.

Definition 5.5 Let A be a finite set and $\Gamma \subseteq \mathscr{R}_A$ a set of finitary relations on A. Then $\mathrm{CSP}_A(\Gamma)$ is the class of all pairs $(X,\mathscr{C})$ such that $(X,A,\mathscr{C})$ is a constraint satisfaction problem where $\mathscr{C} = \{(\mathbf{x}^1,\rho_1),\ldots,(\mathbf{x}^k,\rho_k)\}$ and $\rho_i \in \Gamma$ for all i.

Example 5.6 Let A be a finite set with $|A| \geqslant 2$. Without loss of generality we may assume that $\{0,1\} \subseteq A$. Fix an integer $k \geqslant 2$. For each tuple $\mathbf{a} \in \{0,1\}^k$ let

$$\theta_{\mathbf{a}} = A^k \setminus \{\mathbf{a}\}$$

and let

$$\Theta_k = \{\theta_{\mathbf{a}} : \mathbf{a} \in \{0,1\}^k\}.$$

Example 5.1 suggests that $\theta_{\mathbf{a}}$, where $\mathbf{a} = (a_1,\ldots,a_k)$, consists of all the k-tuples $(b_1,\ldots,b_k) \in \{0,1\}^k$ such that $x_{i1}^{1-a_1} \vee \ldots \vee x_{ik}^{1-a_k}$ evaluates to 1 under the assignment of truth values $x_{i1} = b_1$, …, $x_{ik} = b_k$. Therefore, $\mathrm{CSP}_A(\Theta_k)$ corresponds to the k-SATISFIABILITY problem.

More precisely, it is easy to see that the k-SATISFIABILITY problem is polynomially reducible to $\mathrm{CSP}_A(\Theta_k)$. Take any propositional formula

$$F(x_1,\ldots,x_n) = \bigwedge_{i=1}^{m} (x_{i1}^{\varepsilon_{i1}} \vee \ldots \vee x_{ik}^{\varepsilon_{ik}})$$

in its conjuctive normal form over a set of variables $X = \{x_1,\ldots,x_n\}$. Let $\mathbf{x}^i = (x_{i1},\ldots,x_{ik})$, for each $j \in \{1,\ldots,m\}$ take the constraint $(\mathbf{x}^j,\theta_{\mathbf{a}_j})$ where $\mathbf{a}_j = (1-\varepsilon_{j1},\ldots,1-\varepsilon_{jk})$, and put

$$\mathscr{C}_F = \{(\mathbf{x}^1,\theta_{\mathbf{a}_1}),\ldots,(\mathbf{x}^m,\theta_{\mathbf{a}_m})\}.$$

Then $(X, \mathscr{C}_F) \in \mathrm{CSP}_A(\Theta_k)$ and it is obvious that $(X, \mathscr{C}_F)$ has a solution if and only if F is a satisfiable formula. Since the k-SATISFIABILITY problem is NP-complete for $k \geqslant 3$, this polynomial reduction shows that $\mathrm{CSP}_A(\Theta_k)$ is NP-complete for $k \geqslant 3$.

Example 5.7 An instance of the NOT-ALL-EQUAL 3-SATISFIABILITY consists of a set of triples $\{(x_1, y_1, z_1), \ldots, (x_n, y_n, z_n)\} \subseteq X^3$ such that X is a finite set of variables and $x_i \neq y_i \neq z_i \neq x_i$ for all i. The question is whether there exists a valuation $f : X \to \{0,1\}$ such that $\neg(f(x_i) = f(y_i) = f(z_i))$ for all i, i.e. no triple evaluates to $(0,0,0)$ or $(1,1,1)$. It is a well-known fact that NOT-ALL-EQUAL 3-SATISFIABILITY problem is NP-complete [55]. We shall now provide an interpretation of the problem in terms of CSP.

Let A be a finite set with $|A| \geqslant 2$ and assume that $\{0,1\} \subseteq A$. Consider the relation

$$\beta = \{0,1\}^3 \setminus \{(0,0,0), (1,1,1)\}.$$

Then the NOT-ALL-EQUAL 3-SATISFIABILITY problem is polynomially reducible to $\mathrm{CSP}_A(\{\beta\})$ whence follows that $\mathrm{CSP}_A(\{\beta\})$ is NP-complete.

Equivalently, CSP can be understood as a class of decision problems $(\mathscr{X}, \mathscr{A})$, where $\mathscr{X} = (X, \xi_1, \ldots, \xi_k)$ and $\mathscr{A} = (A, \rho_1, \ldots, \rho_k)$ are finite relational systems of the same type (that is, $\mathrm{ar}(\xi_i) = \mathrm{ar}(\rho_i)$ for all i), and the problem is to decide whether there exists a homomorphism $f : \mathscr{X} \to \mathscr{A}$. Recall that a homomorphism between relational systems $\mathscr{X}$ and $\mathscr{A}$ is a mapping $f : X \to A$ such that $f(\xi_i) \subseteq \rho_i$, for all i.

The two formulations of CSP are equivalent in the following sense:

Lemma 5.8 *For every instance $(X, A, \mathscr{C})$ of the general constraint satisfaction problem there exist finite relational systems $\mathscr{X}_\mathscr{C}$ and $\mathscr{A}_\mathscr{C}$ of the same type such that there is a valuation which satisfies every constraint in $\mathscr{C}$ if and only if there is a homomorphism from $\mathscr{X}_\mathscr{C}$ to $\mathscr{A}_\mathscr{C}$.*

Conversely, for every pair of finite relational systems $\mathscr{X}$ and $\mathscr{A}$ of the same type there exists an instance $(X, A, \mathscr{C}_{\mathscr{X},\mathscr{A}})$ of the general constraint satisfaction problem such that there is a homomorphism from $\mathscr{X}$ to $\mathscr{A}$ if and only if that there is a valuation which satisfies every constraint in $\mathscr{C}_{\mathscr{X},\mathscr{A}}$.

Moreover, an instance of CSP is in P (NP-complete) if and only if its analogon is in P (NP-complete).

Proof. Indeed, take any finite set of constraints $\mathscr{C} = \{(\mathbf{x}^1, \rho_1), \ldots, (\mathbf{x}^k, \rho_k)\}$ and let $\mathscr{A}_\mathscr{C} = (A, \rho_1, \ldots, \rho_k)$ and $\mathscr{X}_\mathscr{C} = (X, \xi_1, \ldots, \xi_k)$, where $\xi_i = \{\mathbf{x}^i\}$ for all i. Then a mapping $f : X \to A$ is a valuation which satisfies every constraint in $\mathscr{C}$ if and only if f is a homomorphism from $\mathscr{X}_\mathscr{C}$ to $\mathscr{A}_\mathscr{C}$.

Conversely, let $\mathscr{X} = (X, \xi_1, \ldots, \xi_k)$ and $\mathscr{A} = (A, \rho_1, \ldots, \rho_k)$ be finite relational systems of the same type and let $\mathscr{C}_{\mathscr{X},\mathscr{A}}$ be the following set of constraints:

$$\mathscr{C}_{\mathscr{X},\mathscr{A}} = \mathscr{C}_1 \cup \ldots \cup \mathscr{C}_k,$$

where

$$\mathscr{C}_i = \{(\mathbf{x}, \rho_i) : \mathbf{x} \in \xi_i\}.$$

Then a mapping $f : X \to A$ is a homomorphism from $\mathscr{X}$ to $\mathscr{A}$ if and only if f is a valuation which satisfies every constraint in $\mathscr{C}_{\mathscr{X},\mathscr{A}}$.

Note that the constructions $\mathscr{C} \mapsto (\mathscr{X}_{\mathscr{C}}, \mathscr{A}_{\mathscr{C}})$ and $(\mathscr{X}, \mathscr{A}) \mapsto \mathscr{C}_{\mathscr{X},\mathscr{A}}$ described above can be realized in polynomial time. □

Let $\mathscr{A} = (A, \Gamma)$ be a finite relational structure (the *template*). Then instead of $\mathrm{CSP}_A(\Gamma)$ it may be more convenient to write $\mathrm{CSP}(\mathscr{A})$. In this parlance, $\mathrm{CSP}(\mathscr{A})$ then denotes the class of all finite relational structures $\mathscr{X}$ of the same type as $\mathscr{A}$ such that there is a homomorphism $\mathscr{X} \to \mathscr{A}$. With a slight abuse of set notation, we might write

$$\mathrm{CSP}(\mathscr{A}) = \{\mathscr{X} : \mathscr{X} \text{ is a finite relational structure of the same type as } \mathscr{A} \text{ and there is a homomorphism } \mathscr{X} \to \mathscr{A}\}$$

The formulation of CSP via homomorphisms sometimes allows for a more compact description of the problem.

Example 5.1bis In order to obtain an analogon of the k-SATISFIABILITY problem in terms of homomorphism of relational stuctures, we apply the algorithm from the proof of of Lemma 5.8. So, $\mathscr{X} = (X, \xi_1, \ldots, \xi_m)$ where $X = \{x_1, \ldots, x_k\}$, $\xi_1 = \ldots = \xi_m = \{(x_1, \ldots, x_k)\}$, and $\mathscr{A} = (\{0,1\}, \rho_1, \ldots, \rho_m)$, where ρ_i's are defined in Example 5.1.

The following examples are more instructive.

Example 5.2bis For an analogon of the k-GRAPHCOLORABILITY problem in terms of homomorphism of relational stuctures it suffices to take $\mathscr{X} = G$ and $\mathscr{A} = K_k$, the complete graph on k vertices (that is, an appropriate representation of these two graphs, such as ε_G in Example 5.3). To see this, it suffices to note that if u and v are adjacent in G, and if $f : G \to K_k$ is a graph homomorphism, then $f(u) \neq f(v)$, since K_k does not have loops.

Example 5.3bis For an analogon of the k-CLIQUE problem just take $\mathscr{X} = K_k$ and $\mathscr{A} = G$.

Example 5.4bis Finally, for an analogon of the HAMILTONIAN problem we have to ensure that the graph homomorphism be injective. Therefore, take a graph $G = (V,E)$, and let $\mathscr{A} = (V, \varepsilon_G, \nu_V)$ where ε_G is the binary relation defined in Example 5.3 and ν_V is the "non-equality" relation defined in Example 5.2. On the other hand, let $\mathscr{X} = (X, \xi, \nu_X)$ where $X = \{1,2,\ldots,n\}$, $n = |V|$ is the number of vertices of G, ν_X is the "non-equality" relation on X and $\xi = \{(1,2),(2,3),\ldots,(n-1,n),(n,1)\}$. Then G has a Hamiltonian cycle if and only if there is a homomorphism $f : \mathscr{X} \to \mathscr{A}$.

5.2 Constraints and clones

In order to describe tractable sets of relations over $A = \{0,1\}$, Schaefer used syntactic properties of propositional formulas representing boolean relations. However, in case of $|A| \geqslant 3$ this method can no longer be used. We therefore need an adequate language in which it is possible to express the properties of sets of relations which are responsible for the complexity of the corresponding constraint satisfaction problems. A useful first step in tackling this problem is to consider what additional relations can be added to a set of relations without changing the complexity of the corresponding problem class. The main result in this section is due to Jeavons and shows that the complexity of the constraint satisfaction problem does not increase if we pass from a set of relations to the relational clone generated by the set of relations [28].

We shall say that a problem is *tractable* if there exists a deterministic polynomial-time algorithm that solves all the instances of that problem. In order to be able to talk about tractability of infinite as well as finite sets of relations, we follow [12] and define the notion of a tractable set of relations in a way that depends on finite subsets only.

Definition 5.9 Let A be a finite set and let $\Gamma \subseteq \mathscr{R}_A$ be finite. We say that Γ is *tractable* if $\mathrm{CSP}_A(\Gamma)$ is tractable. We say that Γ is *NP-complete* if $\mathrm{CSP}_A(\Gamma)$ is NP-complete.

Now, let $\Gamma \subseteq \mathscr{R}_A$ be infinite. We say that Γ is *tractable* if every finite $\Delta \subseteq \Gamma$ is tractable. We say that Γ is *NP-complete* if there is a finite $\Delta \subseteq \Gamma$ which is NP-complete. We say that Γ is *globally tractable* if $\mathrm{CSP}_A(\Gamma)$ is tractable, i.e., every decision problem in $\mathrm{CSP}_A(\Gamma)$ is in P.

Theorem 5.10 (Jeavons 1998, [28]) *Let $\Gamma \subseteq \mathscr{R}_A$ be an arbitrary set of relations (finite or infinite). Then for every finite $\Delta \subseteq \mathrm{Clr}(\Gamma)$ there exists a polynomial-time algorithm which reduces every instance of $\mathrm{CSP}_A(\Delta)$ to an instance of $\mathrm{CSP}_A(\Gamma)$. In*

other words, for every finite $\Delta \subseteq \mathrm{Clr}(\Gamma)$*, the class of problems* $\mathrm{CSP}_A(\Delta)$ *is polynomially reducible to* $\mathrm{CSP}_A(\Gamma)$*.*

Proof. (See [12]) Let $\Delta = \{\theta_1, \ldots, \theta_k\} \subseteq \mathrm{Clr}(\Gamma)$ be a finite set of relations. We know from Theorem 4.52 that for every $\theta_i \in \Delta$ there exists a primitive-positive formula φ_i over Γ such that

$$(a_1, \ldots, a_h) \in \theta_i \quad \text{if and only if} \quad (A, \Gamma) \models \varphi_i[a_1, \ldots, a_h]$$

for all $a_1, \ldots, a_h \in A$. Note that it is not the job of the algorithm we are looking for to find these formulas φ_i. Given a finite template Δ, we use our human ingenuity to find the formulas φ_i which are, then, hard-coded into the algorithm. So, for each θ_i fix a primitive-positive formula φ_i defining θ_i in terms of relations from Γ.

Now, take any instance $(X, \mathscr{C}) \in \mathrm{CSP}_A(\Delta)$ where $X = \{x_1, \ldots, x_n\}$ and $\mathscr{C} = \{(\mathbf{x}^1, \theta_{i_1}), \ldots, (\mathbf{x}^m, \theta_{i_m})\}$. For each $(\mathbf{x}^j, \theta_{i_j}) \in \mathscr{C}$ repeat the following:

- let $\varphi_{i_j}(x_{u_1}, \ldots, x_{u_r}) = \exists y_1, \ldots, y_p(\rho_1(z_1^1, \ldots, z_{l_1}^1) \wedge \ldots \wedge \rho_q(z_1^q, \ldots, z_{l_q}^q))$, where $\rho_t \in \Gamma \cup \{=\}$ for all t and $z_t^s \in \{x_{u_1}, \ldots, x_{u_r}, y_1, \ldots, y_p\}$, be the primitive-positive formula defining θ_{i_j};
- add the auxiliary variables $y_1, \ldots, y_p$ to X (renaming if necessary so that none of them occurs before);
- add the constraints $((z_1^1, \ldots, z_{l_1}^1), \rho_1), \ldots, ((z_1^q, \ldots, z_{l_q}^q), \rho_q)$ to $\mathscr{C}$;
- remove $(\mathbf{x}^j, \theta_{i_j})$ from $\mathscr{C}$.

It can easily be checked that the instance $(X', \mathscr{C}')$ obtained by this procedure is equivalent to $(X, \mathscr{C})$ and belongs to $\mathrm{CSP}_A(\Gamma \cup \{=\})$. Moreover, since all the primitive-positive formulas representating relations from Δ are fixed, this transformation can be carried out in polynomial time. Finally, all constraints of the form $((x,y),=)$ can be eliminated by replacing all occurrences of the variable x with y. This transformation can also be carried out in polynomial time. □

This result reduces the problem of characterizing tractable sets of constraints to the problem of characterizing tractable relational clones:

Corollary 5.11 *Let* $\Gamma \subseteq \mathscr{R}_A$ *be an arbitrary set of relations. Then* Γ *is tractable if and only if* $\mathrm{Clr}(\Gamma)$ *is tractable. Moreover,* Γ *is NP-complete if and only if* $\mathrm{Clr}(\Gamma)$ *is NP-complete.*

We have shown, thus, that in order to analyze the complexity of arbitrary sets of relations it suffices to consider only relational clones. This is not only a considerabe reduction in the sense that, in contrast to arbitraty sets of relations, relational

clones are well understood, but enables us to use the description of relational clones via clones of operations. As we shall see, the tractability of $\mathrm{CSP}_A(\Gamma)$ depends significantly on the structure of $\mathrm{Pol}\,\Gamma$.

5.3 Tidying up

Quite often it is possible to find a solution to a CSP problem by taking a partial solution and extending it to the global solution. Those approaches to solving CSP usually require tidying up the instance we are working with. In this section we describe two such procedures which we need in the sequel.

Let $(X,\mathscr{C})$ be an instance of $\mathrm{CSP}_A(\Gamma)$ where Γ is a relational clone and let $\mathscr{C} = \{(\mathbf{x}^1,\rho_1),\dots,(\mathbf{x}^k,\rho_k)\}$. Fix a linear order on X, say, $X = \{x_1,\dots,x_n\}$ and consider the following algorithm:

(1) for each $x \in X$ do
    if x appears in no tuple $\mathbf{x}^i$ then
        remove x from X;

(2) for each $i \in \{1,\dots,k\}$ do
    if $\mathbf{x}^i = (\dots,x_j,\dots,x_j,\dots)$ then
        introduce a new letter y and put it at the end of X
            (recall that X is linearly ordered)
        remove $(\mathbf{x}^i,\rho_i)$ from $\mathscr{C}$
        add $(\mathbf{y}^i,\rho_i)$ to $\mathscr{C}$, where $\mathbf{y}^i = (\dots,x_j,\dots,y,\dots)$
        add $((x_j,y),\delta_2^{12})$ to $\mathscr{C}$
            (recall that $\delta_2^{12} = \{(x,x) : x \in A\}$)

(3) for each $i \in \{1,\dots,k\}$ do
    let f be a permutation such that $\mathrm{pr}_f(\mathbf{x}^i)$ is sorted w.r.t. the order of X
    remove $(\mathbf{x}^i,\rho_i)$ from $\mathscr{C}$
    add $(\mathrm{pr}_f(\mathbf{x}^i),\mathrm{pr}_f(\rho_i))$ to $\mathscr{C}$

This simple algorithm runs in polynomial time and makes a version of $(X,\mathscr{C})$ we shall refer to as *tidy*. It is tidy in the following sense:

- $(X',\mathscr{C}')$ is an instance of $\mathrm{CSP}_A(\Gamma)$;
- $(X,\mathscr{C})$ has a solution if and only if $(X,\mathscr{C})$ has a solution;
- every letter from X apperas in at least one constraint;
- for every $(\mathbf{x},\rho) \in \mathscr{C}'$, all the letters in $\mathbf{x}$ are distinct and appear in a fixed order.

A *list* of variables from a linearly ordered set $X = \{x_1, \ldots, x_n\}$ is a tuple $(x_{i_1}, \ldots, x_{i_k})$ such that $i_1 < \ldots < i_k$. So, in a "tidy" instance of $\mathrm{CSP}_A(\Gamma)$ every constraint consists of a list of variables, together with a relation from Γ. A list $(x_{i_1}, \ldots, x_{i_k})$ is *contained in* a list $(x_{j_1}, \ldots, x_{j_l})$ if $\{i_1, \ldots, i_k\} \subseteq \{j_1, \ldots, j_l\}$. In that case we write $(x_{i_1}, \ldots, x_{i_k}) \sqsubseteq (x_{j_1}, \ldots, x_{j_l})$.

Let $S = ((x_{j_1}, \ldots, x_{j_l}), \rho)$ be a constraint where $(x_{j_1}, \ldots, x_{j_l})$ is a list, and let $(x_{i_1}, \ldots, x_{i_k})$ be a list such that $(x_{i_1}, \ldots, x_{i_k}) \sqsubseteq (x_{j_1}, \ldots, x_{j_l})$. Then for each p, x_{i_p} appears at precisely one place in the list $(x_{j_1}, \ldots, x_{j_l})$, say at the place m_p (or, more precisely, $x_{j_{m_p}} = x_{i_p}$). The *projection* of S onto $(x_{i_1}, \ldots, x_{i_k})$ is the constraint

$$\mathrm{pr}_{(x_{i_1}, \ldots, x_{i_k})}(S) = ((x_{i_1}, \ldots, x_{i_k}), \mathrm{pr}_{m_1, \ldots, m_k}(\rho)).$$

For lists $\mathbf{x}^1$ and $\mathbf{x}_2$ over X, let $\mathbf{x}^1 \sqcup \mathbf{x}^2$ denote the shortest list $\mathbf{y}$ over X such that $\mathbf{x}^1 \sqsubseteq \mathbf{y}$ and $\mathbf{x}^2 \sqsubseteq \mathbf{y}$, and let $\mathbf{x}^1 \sqcap \mathbf{x}^2$ denote the longest list $\mathbf{y}$ over X such that $\mathbf{y} \sqsubseteq \mathbf{x}^1$ and $\mathbf{y} \sqsubseteq \mathbf{x}^2$.

Let $S_1 = (\mathbf{x}^1, \rho_1)$ and $S_2 = (\mathbf{x}^2, \rho_2)$ be constraints where both $\mathbf{x}^1$ and $\mathbf{x}^2$ are lists and let m be the length of $\mathbf{x}^1 \sqcup \mathbf{x}^2$. Let $\mathbf{x}^1 = (x_1^1, \ldots, x_k^1)$ and $\mathbf{x}^2 = (x_1^2, \ldots, x_l^2)$. Furthermore, let p_j be the position of x_j^1 in $\mathbf{x}^1 \sqcup \mathbf{x}^2$ and let q_j be the position of x_j^2 in $\mathbf{x}^1 \sqcup \mathbf{x}^2$. The *join* of S_1 and S_2 is the constraint

$$S_1 \bowtie S_2 = (\mathbf{x}^1 \sqcup \mathbf{x}^2, \sigma),$$

where

$$\sigma = \{\mathbf{a} \in A^m : \mathrm{pr}_{p_1, \ldots, p_k}(\mathbf{a}) \in \rho_1 \text{ and } \mathrm{pr}_{q_1, \ldots, q_l}(\mathbf{a}) \in \rho_2\}.$$

Example 5.12 Let $X = \{x, y, z, u, v\}$ with the linear order $x \prec y \prec z \prec u \prec v$ and let $A = \{a, b, c, d\}$. Consider the following two constraints (where the list of the variables appears at the top row of the table, while the tuples from the relation apear in the remaining rows):

S_1 :

x	y	z	u
a	a	a	a
a	c	b	d
a	c	d	c

and S_2 :

x	z	v
a	b	a
a	b	b
a	c	b
a	d	d

Then

$S_1 \bowtie S_2$:

x	y	z	u	v
a	c	b	d	a
a	c	b	d	b
a	c	d	c	d

Lemma 5.13 *(a) Let $S_1 = (\mathbf{x}^1, \rho_1)$ and $S_2 = (\mathbf{x}^2, \rho_2)$ be constraints where both $\mathbf{x}^1$ and $\mathbf{x}^2$ are lists. Then $\mathrm{pr}_{\mathbf{x}^1}(S_1 \bowtie S_2) \subseteq \rho_1$ and $\mathrm{pr}_{\mathbf{x}^2}(S_1 \bowtie S_2) \subseteq \rho_2$.*

(b) Let Γ be a relational clone and let $(X, \{S_1, S_2, S_3, \ldots, S_k\})$ be a "tidy" instance of $\mathrm{CSP}_A(\Gamma)$. Then $(X, \{S_1 \bowtie S_2, S_3, \ldots, S_k\}) \in \mathrm{CSP}_A(\Gamma)$.

(c) Let Γ be a relational clone and let $(X, \{S_1, S_2, S_3, \ldots, S_k\})$ be a "tidy" instance of $\mathrm{CSP}_A(\Gamma)$. Then $f : X \to A$ is a solution to $(X, \{S_1, S_2, S_3, \ldots, S_k\})$ if and only if f is a solution to $(X, \{S_1 \bowtie S_2 \bowtie \ldots \bowtie S_k\})$. In particular, $(X, \{S_1, S_2, S_3, \ldots, S_k\})$ has a solution if and only if $S_1 \bowtie S_2 \bowtie \ldots \bowtie S_k = (\mathbf{x}, \rho)$ where ρ is nonempty.

Proof. (a) Obvious.

(b) Let $S_1 = (\mathbf{x}^1, \rho_1)$, $S_2 = (\mathbf{x}^2, \rho_2)$ and let $S_1 \bowtie S_2 = (\mathbf{y}, \sigma)$. It is easy to see that σ can be obrained from ρ_1 and ρ_2 using diagonals, pr and $\times$, so Theorem 4.28 implies that $\sigma \in \Gamma$ since Γ is a relational clone. Therefore, $(X, \{S_1 \bowtie S_2, S_3, \ldots, S_k\})$ is an instance of $\mathrm{CSP}_A(\Gamma)$.

(c) is a straightforward consequence of (a) and the definition of the join of constraints. □

Note that statement (c) in the previous lemma *does not provide* a feasible algorithm for solving CSP in general because it is not clear why computing $S_1 \bowtie S_2 \bowtie \ldots \bowtie S_k$ should take polynomial time in the length of the input!

Another way of transforming an instance of a CSP into a (hopefully) more managable one consists of removing from the constraints those tuples for which we know that cannot contribute to finding a solution. To illustrate the idea, take two constraints $S_1 = (\mathbf{x}^1, \rho_1)$ and $S_2 = (\mathbf{x}^2, \rho_2)$ over the same set of variables X. Take a list $\mathbf{y} = (y_1, \ldots, y_k)$ such that $\mathbf{y} \sqsubseteq \mathbf{x}^1$ and $\mathbf{y} \sqsubseteq \mathbf{x}^2$, let p_j be the position of y_j in $\mathbf{x}^1$ and let q_j be the position of y_j in $\mathbf{x}^2$. Assume now that there is a tuple $\mathbf{a} = (a_1, \ldots, a_k) \in \mathrm{pr}_{p_1,\ldots,p_k}(\rho_1) \setminus \mathrm{pr}_{q_1,\ldots,q_k}(\rho_2)$. Then if there exists a solution $f : X \to A$ to $\{S_1, S_2\}$ then we know for sure that $f(y_1) \neq a_1, \ldots, f(y_k) \neq a_k$, since $\mathbf{a} \notin \mathrm{pr}_{q_1,\ldots,q_k}(\rho_2)$. Therefore, we can remove from ρ_1 all those tuples $\mathbf{z}$ having the property $\mathrm{pr}_{p_1,\ldots,p_k}(\mathbf{z}) = \mathbf{a}$.

Instead of "pruning" constraints $S_1 = (\mathbf{x}^1, \rho_1)$ and $S_2 = (\mathbf{x}^2, \rho_2)$ for an arbitrary list $\mathbf{y}$ satisfying $\mathbf{y} \sqsubseteq \mathbf{x}^1$ and $\mathbf{y} \sqsubseteq \mathbf{x}^2$, it is much more efficient to prune them for maximal such $\mathbf{y}$, which is $\mathbf{x}^1 \sqcap \mathbf{x}^2$. The final observation is that "maximally" pruned S_1 and S_2 can be obtained simply as $\mathrm{pr}_{\mathbf{x}^1}(S_1 \bowtie S_2)$ and $\mathrm{pr}_{\mathbf{x}^2}(S_1 \bowtie S_2)$.

The "pruning" algorithm now takes the following form. Let $(X, \mathscr{C})$ be a tidy instance of $\mathrm{CSP}_A(\Gamma)$ where $\mathscr{C} = \{S_1, \ldots, S_k\}$ and $S_i = (\mathbf{x}^i, \rho_i)$, $i \in \{1, \ldots, k\}$.

(1) repeat
(2) for each $i, j \in \{1, \ldots, k\}$ such that $i < j$ do
(3) $S'_i \leftarrow \mathrm{pr}_{\mathbf{x}^i}(S_i \bowtie S_j)$
(4) $S'_j \leftarrow \mathrm{pr}_{\mathbf{x}^j}(S_i \bowtie S_j)$
(5) remove S_i and S_j from $\mathscr{C}$
(6) add S_i and S_j to $\mathscr{C}$
(7) until no more changes to $\mathscr{C}$

Lemma 5.14 *Let $(X, \mathscr{C})$ be a tidy instance of $\mathrm{CSP}_A(\Gamma)$ and let $(X, \mathscr{C}')$ be the outcome of the "pruning" algorithm. Then $(X, \mathscr{C}')$ be a tidy instance of $\mathrm{CSP}_A(\Gamma)$.*

Proof. It is obvious that $(X, \mathscr{C}')$ is tidy, because the algorithm changes neither X nor the tuples $\mathbf{x}^1, \ldots, \mathbf{x}^k$. Moreover, it is easy to see that $\mathrm{pr}_{\mathbf{x}^i}(S_i \bowtie S_j)$ can be obrained from ρ_i and ρ_j using diagonals, pr and $\times$, so Theorem 4.28 implies that $\mathrm{pr}_{\mathbf{x}^i}(S_i \bowtie S_j) \in \Gamma$ since Γ is a relational clone. Therefore, $(X, \mathscr{C}')$ be a tidy instance of $\mathrm{CSP}_A(\Gamma)$. □

Lemma 5.15 *The "pruning" algorithm runs in polynomial time.*

Proof. It is clear that each of the constructions in steps (3)–(6) takes polynomial time, so the body of the repeat-until loop (lines (2)–(6)) executes in polynomial time. In each pass through the body of the repeat-until loop we remove at least one tuple from one of the relations $\rho_1, \ldots, \rho_k$ (the algorithm stops when no such removal occurs). In the worst case, the algorithm executes the repeat-until loop once for each tuple of each of the relations $\rho_1, \ldots, \rho_k$, and the numer of tuples equals $|\rho_1| + \ldots + |\rho_n|$ is polynomial in the length of the input. Therefore, the entire algorithm runs in polynomial time. □

Each "pruned" instance of $\mathrm{CSP}_A(\Gamma)$ is "consistent" in the following sense (a precise notion of consistency will be introduced later):

Lemma 5.16 *Let $(X, \mathscr{C})$ be an outcome of the "pruning" algorithm, let $S_1 = (\mathbf{x}^1, \rho_1)$ and $S_2 = (\mathbf{x}^2, \rho_2)$ be two constraints in $\mathscr{C}$ and let $\mathbf{y}$ be a list over X such that $\mathbf{y} \sqsubseteq \mathbf{x}^1$ and $\mathbf{y} \sqsubseteq \mathbf{x}^2$. Then $\mathrm{pr}_{\mathbf{y}}(S_1) = \mathrm{pr}_{\mathbf{y}}(S_2)$.*

Proof. Let us start by considering the case $\mathbf{y} = \mathbf{x}^1 \sqcap \mathbf{x}^2$. Let $\mathrm{pr}_{\mathbf{x}^1 \sqcap \mathbf{x}^2}(S_1) = (\mathbf{x}^1 \sqcap \mathbf{x}^2, \rho'_1)$ and $\mathrm{pr}_{\mathbf{x}^1 \sqcap \mathbf{x}^2}(S_2) = (\mathbf{x}^1 \sqcap \mathbf{x}^2, \rho'_2)$ and assume that $\mathrm{pr}_{\mathbf{x}^1 \sqcap \mathbf{x}^2}(S_1) \neq \mathrm{pr}_{\mathbf{x}^1 \sqcap \mathbf{x}^2}(S_2)$. Then $\rho'_1 \neq \rho'_2$ so $\rho'_1 \setminus \rho'_2 \neq \varnothing$ or $\rho'_2 \setminus \rho'_1 \neq \varnothing$. This, however, contradicts the fact that $(X, \mathscr{C})$ is an outcome of the "pruning" algorithm, since the "pruning" algorithm

would not have stopped with $(X, \mathscr{C})$ as a result if there had been more possibilities for "pruning".

Now, let $\mathbf{y}$ be a list over X such that $\mathbf{y} \sqsubseteq \mathbf{x}^1$ and $\mathbf{y} \sqsubseteq \mathbf{x}^2$. Then $\mathbf{y} \sqsubseteq \mathbf{x}^1 \sqcap \mathbf{x}^2$, so starting from $\mathrm{pr}_{\mathbf{x}^1 \sqcap \mathbf{x}^2}(S_1) = \mathrm{pr}_{\mathbf{x}^1 \sqcap \mathbf{x}^2}(S_2)$ and taking $\mathrm{pr}_{\mathbf{y}}$ of the both sides we obtain the claim. □

5.4 Towards the dichotomy

Shaefer proved in 1978 [55] that every $\mathrm{CSP}_{\{0,1\}}(\Gamma)$ is either tractable or NP-complete. In 1993 it was conjectured by Feder and Vardi that every $\mathrm{CSP}_A(\Gamma)$ where A is a finite set is either tractable or NP-complete. This is called the *Dichotomy Conjecture*:

The Dichotomy Conjecture (Feder, Vardi 1993, [24]). *For every finite A, every $\mathrm{CSP}_A(\Gamma)$ is either tractable or NP-complete.*

We shall now demonstrate one possibility towards the proof of the conjecture. Let Γ be a relational clone such that $\Gamma \neq \mathscr{R}_A$. Then $\mathrm{Pol}\,\Gamma \neq \Pi_A$ and according to Theorems 4.33 and 4.36 one of the following cases arises:

(1a) $\mathrm{Pol}\,\Gamma$ contains a constant unary operation;

(1b) $\mathrm{Pol}\,\Gamma$ contains essentially unary operations *only*, none of which is a constant;

(2) $\mathrm{Pol}\,\Gamma$ contains a binary idempotent operation which is not a projection;

(3) $\mathrm{Pol}\,\Gamma$ contains a majority operation;

(4) $\mathrm{Pol}\,\Gamma$ contains a minority operation; or

(5) $\mathrm{Pol}\,\Gamma$ contains projections and semiprojections only.

5.4.1 Constants (Case (1a))

Proposition 5.17 (Jeavons 1998, [28]) *If $\mathrm{Pol}\,\Gamma$ contains a constant unary operation then $\mathrm{CSP}_A(\Gamma)$ can be solved in polynomial time.*

Proof. Let c_a be a constant unary operation such that $c_a \in \mathrm{Pol}\,\Gamma$. Then every nonempty relation in Γ contains a tuple of the form $(a, a, \ldots, a)$. Take any instance $(X, \mathscr{C}) \in \mathrm{CSP}_A(\Gamma)$ and let $\mathscr{C} = \{(\mathbf{x}^1, \rho_1), \ldots, (\mathbf{x}^k, \rho_k)\}$. If there is an i such that $\rho_i = \varnothing$ then the instance of the problem has no solutions. Otherwise, c_a is a solution. This can be decided in polynomial time. □

5.4.2 Semiprojections (Case (5))

Proposition 5.18 (Jeavons 1998, [28]) *If* $\operatorname{Pol}\Gamma$ *contains projections and semiprojections only then* $\mathrm{CSP}_A(\Gamma)$ *is NP-complete.*

Proof. Without loss of generality we can assume that $\{0,1\} \subseteq A$ and recall the definition of Θ_3 from Example 5.6: $\Theta_3 = \{\theta_{\mathbf{a}} : \mathbf{a} \in \{0,1\}^3\}$, where $\theta_{\mathbf{a}} = A^3 \setminus \{\mathbf{a}\}$. It is easy to see that every $\theta_{\mathbf{a}}$ is invariant under projections and semiprojections. Since $\operatorname{Pol}\Gamma$ contains projections and semiprojections only, it follows that $\operatorname{Pol}\Gamma \subseteq \operatorname{Pol}\Theta_3$, i.e., $\Theta_3 \subseteq \operatorname{Clr}(\Gamma)$. Theorem 5.10 now yields that $\mathrm{CSP}_A(\Theta_3)$ is polynomially reducible to $\mathrm{CSP}_A(\Gamma)$. However, Example 5.6 shows that the 3-SATISFIABILITY problem is polynomially reducible to $\mathrm{CSP}_A(\Theta_3)$. Since the 3-SATISFIABILITY problem is NP-complete, it follows that $\mathrm{CSP}_A(\Gamma)$ is also NP-complete. □

5.4.3 Essentially unary nonconstant operations (Case (1b))

Lemma 5.19 *Let* $A = \{a_1, \ldots, a_n\}$.

(a) *Every operation in a clone* C *of operations on* A *is an essentially unary operation if and only if* $\omega_4 = \delta_4^{12|3|4} \cup \delta_4^{1|2|34} \in \operatorname{Inv} C$.

(b) *If every operation in* $\operatorname{Pol}\Gamma$ *is essentially unary, then* $\operatorname{Clr}(\Gamma) = \operatorname{Clr}(\{\rho, \omega_4\})$ *where* ω_4 *is defined in* (a) *and*

$$\rho = \{(f(a_1), \ldots, f(a_n)) : f \in \operatorname{End}\Gamma\}.$$

In particular, $\operatorname{Clr}(\Gamma)$ *is finitely generated.*

Proof. (a) The implication $(\Rightarrow)$ is trivial. Let us show $(\Leftarrow)$.

Let $f \in C$ be an operation that is not essentially unary, let $k = \operatorname{ar}(f) \geqslant 2$ and let $\theta = \delta_4^{12|3|4} \cup \delta_4^{1|2|34}$. Without loss of generality we can assume that f depends on the first two arguments. Then there exist a_1, a_1', b_2, …, b_k, d_1, c_2, c_2', d_3, …, $d_k \in A$ such that

$$p = f(a_1, b_2, b_3, \ldots, b_k) \neq f(a_1', b_2, b_3, \ldots, b_k) = p'$$

and

$$q = f(d_1, c_2, d_3, \ldots, d_k) \neq f(d_1, c_2', d_3, \ldots, d_k) = q'.$$

Then

$$\begin{array}{cccccccc}
a_1 & b_2 & b_3 & \ldots & b_k & \overset{f}{\mapsto} & p \\
a_1' & b_2 & b_3 & \ldots & b_k & \overset{f}{\mapsto} & p' \\
d_1 & c_2 & d_3 & \ldots & d_k & \overset{f}{\mapsto} & q \\
d_1 & c_2' & d_3 & \ldots & d_k & \overset{f}{\mapsto} & q' \\
\cap\!\!\!| & \cap\!\!\!| & \cap\!\!\!| & \cap\!\!\!| & \cap\!\!\!| & & \not\cap\!\!\!| \\
\theta & \theta & \theta & \theta & \theta & & \theta
\end{array}$$

Therefore, f does not preserve θ.

(b) It is easy to see that $\mathrm{Pol}\,\Gamma = \mathrm{Pol}\{\omega_4, \rho\}$, whence $\mathrm{Clr}(\Gamma) = \mathrm{Clr}(\{\omega_4, \rho\})$. □

Lemma 5.20 (Jeavons 1998, [28]) *Let $\Gamma = \{\rho_1, \ldots, \rho_k\}$ be a finite set of relations on A and let $f \in \mathrm{End}\,\Gamma$. Let $B = f(A)$ and let Δ be the following set of relations on B:*

$$\Delta = \{f(\rho_1), \ldots, f(\rho_k)\}.$$

Then $\mathrm{CSP}_A(\Gamma)$ is polynomially reducible to $\mathrm{CSP}_B(\Delta)$ and, vice versa, $\mathrm{CSP}_B(\Delta)$ is polynomially reducible to $\mathrm{CSP}_A(\Gamma)$.

Proof. Let $(X, \mathscr{C})$ be an instance of $\mathrm{CSP}_A(\Gamma)$ where $\mathscr{C} = \{(\mathbf{x}^1, \rho_{i_1}), \ldots, (\mathbf{x}^l, \rho_{i_l})\}$. Then $\mathscr{C}' = \{(\mathbf{x}^1, f(\rho_{i_1})), \ldots, (\mathbf{x}^l, f(\rho_{i_l}))\}$ is an instance of $\mathrm{CSP}_B(\Delta)$ and it is easy to see that if $h : X \to A$ is a solution to $(X, \mathscr{C})$ then $f \circ h$ is a solution to $(X, \mathscr{C}')$. On the other hand, if $h : X \to B$ is a solution to $(X, \mathscr{C}')$ then the same function solves $(X, \mathscr{C})$, since $f(\rho) \subseteq \rho$ for every $\rho \in \Gamma$.

Conversely, let $(X, \mathscr{C}')$ be an instance of $\mathrm{CSP}_B(\Delta)$ where $\mathscr{C}' = \{(\mathbf{x}^1, \theta_1), \ldots, (\mathbf{x}^l, \theta_l)\}$. For each j find a $\rho_{i_j} \in \Gamma$ such that $f(\rho_{i_j}) = \theta_j$ (since Γ is finite this can be achieved by a straightforward polynomial-time algorithm: for each $\rho \in \Gamma$ test whether $f(\rho) = \theta_j$) and put $\mathscr{C} = \{(\mathbf{x}^1, \rho_{i_1}), \ldots, (\mathbf{x}^l, \rho_{i_l})\}$. Clearly, $(X, \mathscr{C})$ is an instance of $\mathrm{CSP}_A(\Gamma)$ and $(X, \mathscr{C}')$ has a solution if and only if $(X, \mathscr{C})$ has a solution. □

Proposition 5.21 (Jeavons 1998, [28]) *If $\mathrm{Pol}\,\Gamma$ contains essentially unary nonconstant operations only, then $\mathrm{CSP}_A(\Gamma)$ is NP-complete.*

Proof. In this proof we work with clones on two sets, A and B, so we shall have to write $\mathrm{Pol}_A\,\Gamma$ and $\mathrm{Pol}_B\,\Gamma$ to distinguish between clones of operations on A and clones of operations on B. Simliarly, we shall write $\omega_4(A)$ and $\omega_4(B)$ (see Lemma 5.19) to distinguish between the two relations which are constucted in the same fashion, but on distinct sets.

Since $\mathrm{Pol}_A\,\Gamma$ contains essentially unary operations only, Lemma 5.19 (b) yields that $\mathrm{Clr}_A(\Gamma) = \mathrm{Clr}_A(\{\rho, \omega_4(A)\})$, where ρ and $\omega_4(A)$ are defined in Lemma 5.19. Moreover, we may safely restrict our attention to the unary operations in the monoid $M = \mathrm{End}_A\,\Gamma = \mathrm{End}_A\{\rho, \omega_4(A)\}$.

Let $q = \min\{|f(A)| : f \in M\}$ and let $g \in M$ be the unary operation that achieves the minimum: $|g(A)| = q$. Since there are no constant maps in M we have $q \geqslant 2$. Let $B = g(A)$ and

$$\Delta = \{g(\rho), g(\omega_4(A))\}.$$

Clearly, $\Delta \subseteq \mathscr{R}_B$. Since $g(\omega_4(A)) = \omega_4(B)$, Lemma 5.19 (a) ensures that $\mathrm{Pol}_B\,\Delta$ consists of essentially unary operations only.

Let us first show that $\mathrm{End}_B\Delta$ contains no constant operations. Assume, to the contrary, that $c_b \in \mathrm{End}_B\Delta$ for some $c \in B$. Then c_b preserves $g(\rho)$, whence follows that $(b,\dots,b) \in g(\rho)$. On the other hand, g preserves ρ, whence $g(\rho) \subseteq \rho$. Therefore, $(b,\dots,b) \in \rho$, which contradicts the assumption that $\mathrm{Pol}_A\Gamma$ contains no constant operations.

Next, let us show that every operation in $\mathrm{End}_B\Delta$ is a permutation. Assume now that there exists an $h \in \mathrm{End}_B\Delta$ which is not a permutation. Then $h \circ g \in \mathrm{End}_A\{\rho, \omega_4(A)\} = M$. On the other hand $|h(B)| < |B| = q$ since h is not a permutation. This contradicts the choice of g.

Let us now show that $\mathrm{CSP}_B(\Delta)$ is NP-complete. Lemma 5.20 then yields that $\mathrm{CSP}_A(\Gamma)$ is also NP-complete since $\mathrm{CSP}_B(\Delta)$ is polynomially reducible to $\mathrm{CSP}_A(\Gamma)$.

Assume, first, that $q = 2$. Without loss of generality we can assume that $B = \{0,1\}$. Since every operation in $\mathrm{Pol}_B\Delta$ is essentially unary and every operation in $\mathrm{End}_B\Delta$ is a permutation, it follows that $\beta \in \mathrm{Clr}_B(\Delta)$, where

$$\beta = \{0,1\}^3 \setminus \{(0,0,0),(1,1,1)\}$$

(see Example 5.7). Then Theorem 5.10 ensures that $\mathrm{CSP}_B(\{\beta\})$ is polynomially reducible to $\mathrm{CSP}_B(\Delta)$, and we know from Example 5.7 that $\mathrm{CSP}_B(\{\beta\})$ is NP-complete. Therefore, $\mathrm{CSP}_B(\Delta)$ is NP-complete.

Assume, now, that $q \geqslant 3$. Since every operation in $\mathrm{Pol}_B\Delta$ is essentially unary and every operation in $\mathrm{End}_B\Delta$ is a permutation, it follows that $\nu_B \in \mathrm{Clr}_B(\Delta)$, where

$$\nu_B = \{(x,y) \in B : x \neq y\}$$

(see Example 5.2). Then Theorem 5.10 ensures that $\mathrm{CSP}_B(\{\nu_B\})$ is polynomially reducible to $\mathrm{CSP}_B(\Delta)$. On the other hand, we know from Example 5.2 that $\mathrm{CSP}_B(\{\nu_B\})$ corresponds to the q-GRAPHCOLORABILITY problem, which is NP-complete for $q \geqslant 3$ (which is the case). Therefore, $\mathrm{CSP}_B(\Delta)$ is NP-complete. □

5.4.4 Binary idempotent operations (Case (2))

Proposition 5.22 (Jeavons 1998, [28]) *If* $\mathrm{Pol}\Gamma$ *contains a semilattice operation then* $\mathrm{CSP}_A(\Gamma)$ *is tractable.*

Proof. Let $\wedge \in \mathrm{Pol}\Gamma$ be a semilattice operation on A. Take any instance of $\mathrm{CSP}_A(\Gamma)$, tidy it up, "prune" it (see Section 5.3) and denote the outcome by $(X,\mathscr{C})$. As we have seen in Section 5.3, these two procedures execute in polynomial time.

If $\mathscr{C}$ contains an empty constraint, i.e. a constraint of the form $(\mathbf{x},\varnothing)$, then the original problem has no solutions. Assume, now, that every constraint in $\mathscr{C}$ is nonempty and let us show that in this case $(X,\mathscr{C})$ has a solution.

Let $z \in X$ be an arbitrary variable, and let $(\mathbf{x}, \rho)$ be an arbitrary constraint in $\mathscr{C}$. Let us denote $\mathrm{pr}_z(\mathbf{x}, \rho)$ by (z, D_z). Clearly, $D_z \neq \varnothing$. Moreover, since ρ is invariant under $\wedge$, it follows that D_z is also invariant under $\wedge$, whence follows that $\bigwedge D_z \in D_z$. It is also important to note that D_z does not depend on the constraint $(\mathbf{x}, \rho)$: according to Lemma 5.16, for every pair of constraints $(\mathbf{x}, \rho), (\mathbf{y}, \sigma) \in \mathscr{C}$ such that $z \sqsubseteq \mathbf{x}$ and $z \sqsubseteq \mathbf{y}$ we have $\mathrm{pr}_z(\mathbf{x}, \rho) = \mathrm{pr}_z(\mathbf{y}, \sigma)$.

Define $f : X \to A$ by $f(z) = \bigwedge D_z$ and let us show that f is a solution to $(X, \mathscr{C})$. Take any $((y_1, \ldots, y_k), \sigma) \in \mathscr{C}$. Since $\mathrm{pr}_{y_i}(S) = (y_i, D_{y_i})$ and $f(y_i) = \bigwedge D_{y_i} \in D_{y_i}$ for all i, it follows that for every i there exists a tuple $\mathbf{a}_i \in \sigma$ such that $\mathrm{pr}_i(\mathbf{a}_i) = \bigwedge D_{y_i} = f(y_i)$. Then

$$\mathbf{a}_1 \wedge \ldots \wedge \mathbf{a}_k = (\bigwedge D_{y_1}, \ldots, \bigwedge D_{y_k}) = (f(y_1), \ldots, f(y_k)).$$

But, σ is invariant under $\wedge$, so $\mathbf{a}_1 \wedge \ldots \wedge \mathbf{a}_k \in \sigma$. This shows that f is a solution to every constraint in $\mathscr{C}$. $\square$

Bibliography

[1] Adámek J., Herrlich, H., Strecker, G. E., *Abstract and concrete categories*, Jown Wiley & Sons, New York, 1990

[2] Aichinger E., Mašulović D., Pöschel R., Wilson J. S., *Completeness for concrete near-rings*, Journal of Algebra 279 (2004), 61–78

[3] Baker K. A., Pixley, A. F., *Polynomial interpolation and the Chinese remainder theorem for algebraic systems*, Math. Z. 143 (1975), 165–174

[4] Bergman, C., *Categorical equivalence of algebras with a majority term*, Algebra Universalis 40 (1998), 149–175

[5] Bergman C.,Berman J., *Algorithms for categorical equivalence*, Mathematical Structures in Computer Science 8 (1998), 1–15

[6] Bergman C.,Berman J., *Morita equivalence of almost-primal clones*, Journal of Pure and Applied Algebra 108 (1996), 175–201

[7] Bergman C., Juedes D., Slutzki G., *Computational complexity of term-equivalence*, Internat. J. Algebra Comput., 9 (1999), 113–128

[8] Bondarčuk V. G., Kalužnin L. A., Kotov V. N., Romov B. A., *Galois Theory for Post Algebras I–II*, Cybernetics 5(1969), 243–252, 531–539

[9] Börner, F., Pöschel, R., *Clones of operations on binary relations*, Contributions to General Algebra, 7 (1991), 50–70.

[10] Bredikhin, D. A., *On clones generated by primitive-positive operations of Tarski's relation algebras*, Algebra Universalis 38 (1997), 165–174.

[11] Bulatov A., A dichotomy theorem for constraints on a three-element set, in Proceedings 43rd IEEE Symposium on Foundations of Computer Science (FOCS'02), Vancouver, Canada, 2002, 649–658.

[12] Bulatov A., Jeavons P., Krokhin A., Classifying the complexity of constraints using finite algebras, SIAM J. Comput. Vol 34, No 5 (2005), 720–742

[13] Burle, G. A.: *The classes of k-valued logics containing all one-variable functions* (Russian), Diskret. Analiz 10(1967), 3–7

[14] Burris S., Sankappanavar H. P., *A Course in Universal Algebra*, Springer Verlag, New York, 1981

[15] Burris S., Sankappanavar H. P., *Bevezetés az univerzális algebrába*, Takönyvkiadó, Budapest 1988 (Hungarian translation of [14] by B. Csákány)

[16] Burris S., Willard R., *Finitely Many Primitive Positive Clones*, Proceedings of the American Mathematical Society, Vol. 101, No. 3, November 1987, 427–430

[17] Csákány, B., *Homogeneous algebras are functionally complete*, Algebra Universalis 11 (1980), 149–158

[18] Csákány B.: *Completeness in coalgebras*, Acta Sci. Math. 48(1985), 75–84

[19] Clark, D. M., Davey, B. A., *Natural Dualities for the Working Algebraist*, Cambridge studies in advanced mathematics 57, Cambridge Unviersity Press, Cambridge 1998

[20] Denecke K., Lüders O., *Categorical equivalence of varieties and invariant relations*, Algebra universalis 46 (2001), 105–118

[21] Dixon, J. D., Mortimer, B.: *Permutation Groups*, Springer, 1996

[22] Drbohlav K.: *On quasicovarieties*, Acta Fac. Rerum Natur. Univ. Comenian., Math. Mimoriadne Čislo (1971), 17–20

[23] Duffin, R. J., *Topology of series-parallel networks*, J. Math. Anal. Appl. 10 (1965), 303–319.

[24] Feder T., Vardi M. Y., *Monotone monadic SNP and constraint satisfaction*, Proceedings 25th ACM Symposium on the Theory of Computing (STOC), 1993, 612–622

[25] Geiger D., *Closed Systems of Functions and Predicates*, Pacific J. Math., 27 (1968), 95–100

[26] Gierz G., *Morita equivalence of quasi-primal algebras and sheaves*, Algebra Universalis 35 (1996), 570–576

[27] Hu T. K., *Duality for primal algebra theory*, Math. Z. 110 (1969), 180Ű-198

[28] Jeavons P., On the algebraic structure of combinatorial problems, Theoretical Computer Science, 200(1998), 185–204.

[29] Jónsson, B., *The theory of binary relations*, Colloq. Math. Soc., János Bolyai 54, Algebraic Logic, Budapest, Hungary (1988), 245–292.

[30] Kaarli K., Primitivity and simplicity of non-zerosymmetric near-rings, *Communications in Algebra* **26** (1998), 3691–3708.

[31] Knoebel A., *The equational classes generated by single functionally precomplete algebras*, Mem. Amer. Math. Soc. 57 (1985), no. 332, v+83 pp.

[32] Kuznecov, A. V., *Structures with closure and criteria of functional copleteness*, Usp. Mat. Nauk. 16, 2(98) (1961), 201–202 (Russian)

[33] D. Lau, *Die maximalen Klassen von* $\bigcap_{a\in Q}\mathrm{Pol}_k\{a\}$ *für* $Q \subseteq E_k$ *(Ein Kriterium für endliche semi-primale Algebren mit nur trivialen Unteralgebren)*, Rostok. Math. Kolloq. 48 (1995), 27–46

[34] Liebeck, M. W., Praeger, C. E., Saxl J.: *The classification of the maximal subgroups of the finite alternating and symmetric groups*, J. Algebra 111(1987), 365–383

[35] Lyndon, R., *Identities in two-valued calculi*, Trans. Amer. Math. Soc. 71 (1951), 457-465.

[36] Mašulović D.: *The lattice of clones of co-operations*, PhD Thesis, University of Novi Sad, Yugoslavia, 1999 (in Serbian)

[37] Mašulović D., *On dualizing clones as Lawvere theories*, International Journal of Algebra and Computation 16 (2006), 657–687

[38] Mašulović D., Pöschel R., *On the structure of some Tarski clones*, Acta Sci. Math. Szeged 70 (2004), 455–471

[39] McKenzie R., *An algebraic version of categorical equivalence for varieties and more general algebraic theories*, in: A. Ursini and P. Agliano (editors), Logic and Algebra, vol. 180 of Lecture Notes in Pure and Applied Mathematics, Marcel Dekker, 1996, 211-Ű243

[40] McKenzie R. N., McNulty G. F., Taylor W. F., *Algebras, Lattices, Varietes, Vol. I*, Wadsworth & Brooks, 1987

[41] MacLane S., *Categories for the Working Mathematician*, Springer-Verlag, New York, 1972.

[42] Montanari U., Networks of constraints: Fundamental properties and applications to picture processing, Information Sciences, 7 (1974), 95–132.

[43] Murskiĭ V. L., *The existence in three-valued logic of a closed class with a finite basis having no finite complete system of identities*, (in Russian), Dokl. Akad. Nauk SSSR, 163(1965), 815–81

[44] Pilz G., *Near-Rings*, North-Holland Mathematics Studies 23, (revised edition), North-Holland Publishing Company, 1983.

[45] Pixley, A. F., *The ternary discriminator function in universal algebra*, Math. Ann. 191 (1971) 167–180

[46] Pöschel R., *Concrete representation of algebraic structures and a general Galois theory*, in: Contributions to general algebra (Proc. Klagenfurt Conf. 1978), Verlag Joh. Heyn, Klagenfurt, 1979

[47] Pöschel, R., *Untersuchungen von S-Ringen, insbesondere im Gruppenring von p-Gruppen*, Math. Nachr. 60 (1974), 1–27.

[48] Pöschel R., Kalužnin L. A., *Funktionen- und Relationenalgebren, Ein Kaptel der Diskreten Mathematik*, Veb Deutscher Verlag der Wissenschaften, Berlin 1979

[49] Post E., *The Two-valued Iterative Systems of Mathematical Logic*, Ann. Math. Studies 5, Princeton, 1941

[50] Quackenbush R. W., *A New Proof of Rosenberg's Primal Algebra Characterization Theorem*, Finite Algebras and Multiple-valued Logic (Proc. Conf. Szeged 1979), Colloq. Math. Soc. J. Bolyai, 28, North-Holland, Amsterdam, 1981, 603–634

[51] Rosenberg I. G., *Über die funktionale Vollständigkeit in den mehrwertigen Logiken (Struktur der Funktionen von mehreren Veränderlichen auf endlichen Mengen)*, Rozpravy Československe Akad. Věd. Řada Mat. Přírod. Věd. 80 (1970), 3–93 (English translation: *Completeness Properties of Multiple-valued Logic Algebras*, Computer Science and Multiple-valued Logic; Theory and Applications (Rine D. C., ed.), North-Holland, Amsterdam, 1977, 144–186

[52] Rosenberg I. G., *Minimal clones I: the five types*, Lectures in Universal Algebra (Proc. Conf. Szeged 1983), Colloq. Math. Soc. Janos Bolyai 43, North-Holland 1986, 405–427

[53] Scott S. D., Involution near-rings, *Proc. Edinburgh Math. Soc.* (2), 22, **3** (1979), 241–245.

[54] Scott S. D., Transformation near-rings generated by a unit of order three, *Algebra Colloq.* 4, **4** (1997), 371–392.

[55] Schaefer T., The complexity of satisfiability problems, in Proceedings 10th ACM Symposium on Theory of Computing, STOC'78, 1978, 216–226.

[56] Sheffer H. M., *A set of five independent postulates for Boolean algebras with applications to logical constants*, Trans. Amer. Math. Soc. 14 (1913), 481–488

[57] Słupecki, J., *Kriterium pełnosci wielowartoşciewych systemow logiki zdań*, C. R. Seanc. Soc. Sci. Varsovie 32 (1939), 102–109 (English translation: Studia logica 30 (1972), 153–157)

[58] Świerczkowski, S., *Algebras which are independently generated by every n-elements*, Fund. Math. 49 (1960), 93–104

[59] Szábo, L., *Concrete representation of related structures of iniversal algebras I*, Acta Sci. Math. 40 (1978), 175–184

[60] Székely Z.: *On maximal clones of co-operations*, Acta Sci. Math. 53(1989), 43–50

[61] Szendrei Á., *Clones in Universal Algebra*, Les Presses de l' Université de Montréal, 1976

[62] Tarski, A., *On the calculus of relations*, J. Symbolic Logic, 6 (3), 1941, 73–89.

[63] Tarski, A., Givant, S. R., *A formalization of set theory without variables*, Colloquium Publications, vol. 41, American Mathematical Society, Providence, R.I., 1987

[64] Webb, D. L., *Generation of any n-valued logic by one binary operator*, Proc. Nat. Acad. Sci. 21 (1935), 252–254

[65] Wielandt, H., *Finite Permutation Groups*, Academic Press, New York and London, 1964.

[66] Yablonskiĭ, S. V., *Functional constructions in k-valued logic*, Trudy Mat. Inst. Steklov 51 (1958), 5–142 (Russian)

[67] Yanov, Yu. I., Muchnik A. A., *On the existence of k-valued closed classes that have no bases*, Dokl. Akad. Nauk SSSR 127 (1959), 44-46 (Russian)

[68] Zádori L., *Relational sets and categorical equivalence of algebras*, International Journal of Algebra and Computation 7 (1997), 561–576