Hinweise:
1.) Teil deiner Aufgabe ist es, Zwecks Diskussion des Stoffes und Vertiefung des Verständnisses, Kontakte zu deinen Kommiliton*innen aufzubauen und die Hilfen durch die Hiwis in Anspruch zu nehmen.
2.) Du wirst feststellen, dass Notationen hier und in den Vorlesungen nicht notwendig übereinstimmen, ebenso wie die Reihenfolge im Aufbau. Was wichtig ist, sind die Konzepte und ihre Inhalte. Diese müssen verstanden und eingeübt werden und zwar von möglichst vielen Standpunkten aus.

1.) Grundbegriffe: Elemente und Teilmengen

Aufgaben: 5

```
> restart;
```

Definition endlicher Mengen durch Aufzählung der Elemente, Teilmengen

Kommentar:
Das Mengenkonzept ist in Maple hauptsächlich für endliche Mengen realisiert, weshalb wir uns weitgehend auf diesen Fall beschränken wollen. Es versteht sich von selbst, dass von Seiten der Mathematik viele Dinge auch für unendliche Mengen realisiert werden können.

MAPLE u. MATH: Die Aufzählung der Elemente einer **Menge** ist eine gängige Art, Mengen zu definieren. Die Aufzählung steht zwischen geschweiften Klammern und die einzelnen Elemente sind durch Kommata getrennt. Wiederholungen und die Reihenfolge der Elemente werden ignoriert:

```
> M:={1,4,5,3,7};
  N:={7,7,4,5,4,3,1};
  evalb(M=N);
```

$$M := \{1, 3, 4, 5, 7\}$$

$$N := \{1, 3, 4, 5, 7\}$$

$$true \qquad \textbf{(1.1.1)}$$

Oft erlaubt uns eine definierende Eigenschaft, die Menge in etwas kompakterer Form aufzuschreiben. So können wir $A := \{1, 3, 4, 5\}$ auch schreiben als $A = \{x \in M \mid x < 7\}$. (**DENKANSTOSS**: Warum?) Dies lässt sich so realisieren:

```
> A:={1,3,4,5};
  B := select( x-> evalb(x<7), M );
  evalb( A=B );
```

$$A := \{1, 3, 4, 5\}$$

$$B := \{1, 3, 4, 5\}$$

$$true \qquad \textbf{(1.1.2)}$$

MAPLE: Die Mengenschreibweise erlaubt es, sehr schnell mehrfache Folgenglieder wegzustreichen; wir bezahlen jedoch unter Umständen mit einer

Umordnung. Der Übergang von einer Menge zu einer Folge geschieht in Maple zum Beispiel so:

```
> N;
```

$$\{1, 3, 4, 5, 7\} \quad \textbf{(1.1.3)}$$

```
> s:=op(N);
```

$$s := 1, 3, 4, 5, 7 \quad \textbf{(1.1.4)}$$

```
> s[3];
```

$$4 \quad \textbf{(1.1.5)}$$

MAPLE: Bei **Folgen** ist eine Reihenfolge festgelegt, sodass wir das erste, zweite oder allgemein k-te Glied ansprechen können. Es gibt vorgegebene Prozeduren, die Folgen erzeugen:

```
> s:=seq(i^2,i=1..5);
```

$$s := 1, 4, 9, 16, 25 \quad \textbf{(1.1.6)}$$

```
> s[2];
```

$$4 \quad \textbf{(1.1.7)}$$

```
> {s};
```

$$\{1, 4, 9, 16, 25\} \quad \textbf{(1.1.8)}$$

DENKANSTOSS: Wie erzeugt man in Maple sehr schnell die Menge der natürlichen Zahlen von 1 bis 100?

MAPLE: Man kann aus mehreren endlichen Folgen eine neue Folge machen, indem man sie durch ein Komma getrennt hintereinander schreibt.

```
> t:=$7..12; s,t; t,s; {s,t};
```

$$t := 7, 8, 9, 10, 11, 12$$

$$1, 4, 9, 16, 25, 7, 8, 9, 10, 11, 12$$

$$7, 8, 9, 10, 11, 12, 1, 4, 9, 16, 25$$

$$\{1, 4, 7, 8, 9, 10, 11, 12, 16, 25\} \quad \textbf{(1.1.9)}$$

MAPLE u. MATH: Die Elemente einer Menge brauchen keine Zahlen zu sein; sie müssen aber wohlunterscheidbare Objekte sein. Etwas verwirrend wird es, wenn die Elemente ihrerseits Mengen sind:

```
> S:={1,M};
```

$$S := \{1, \{1, 3, 4, 5, 7\}\} \quad \textbf{(1.1.10)}$$

MAPLE: Die Anzahl der Elemente einer Menge bestimmt man so:

```
> nops(S);
  nops(M);
```

$$2$$

$$5 \quad \textbf{(1.1.11)}$$

ÜBUNG [01]:

Wieviele *verschiedene* Folgenglieder hat eine Folge mit 82 Folgengliedern, deren *i*-tes Glied von der Form $(i-41)^4+(i-41)^2+1$ ist - mit $1 \le i \le 82$?
Hinweis: Erstelle eine Menge mit den Werten der Folgenglieder, benutze dazu wie oben den Befehl **seq**.

MAPLE u. MATH: Hier sind die Befehle, die **Mitgliedschaft** in einer Menge ausdrücken bzw. auswerten:

```
> 1 in M;
```

$$1 \in \{1, 3, 4, 5, 7\} \qquad \textbf{(1.1.12)}$$

```
> evalb( 1 in M );
```

true **(1.1.13)**

```
> member(1,M);
```

true **(1.1.14)**

```
> member(M,S);
```

true **(1.1.15)**

MATH: Eine Menge M_1 heißt **Teilmenge** der Menge M_2, wenn jedes Element von M_1 auch Element von M_2 ist.

Notation: $M_1 \subseteq M_2$.

Eine Umformulierung, welche näher an der Aussagenlogik liegt, lautet wie folgt:

Für jedes Element m der Menge M_1 gilt, dass m auch Element der Menge M_2 ist.

$\forall\, m : m \in M_1 \Rightarrow m \in M_2$.

Insbesondere ist die leere Menge (mathematisches Symbol $\varnothing$, Maplesymbol **{}**), die keine Elemente enthält, Teilmenge jeder anderen Menge:

$\varnothing \subseteq M$ für alle Mengen M.

```
> {} subset M;
```

true **(1.1.16)**

```
> M subset M;
```

true **(1.1.17)**

```
> {$0..10} subset M;
```

false **(1.1.18)**

ÜBUNG [02]:

Benutze die aussagenlogische Definition der Teilmengenoperation, um nur mit der Elementoperation **in** nachzurechnen, dass
1) $\{0,\ldots, 10\}$ nicht Teilmenge von M ist.
2) M Teilmenge von $\{0,\ldots, 10\}$ ist.
ist.

Hinweis: Der Befehl **subset** soll ausdrücklich *nicht* verwendet werden.

```
> M;
  {$0..10};
```

$$\{1, 3, 4, 5, 7\}$$

$$\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\} \quad \textbf{(1.1.19)}$$

Teilmengenarithmetik

MATH: Im Folgenden gehen wir immer davon aus, dass alle vorkommenden Mengen Teilmenge einer (nicht immer genau spezifizierten) Gesamtmenge M sind. Der Teilmengenbeziehung entspricht die aussagenlogische Verknüpfung der Implikation, so wie der Gleichheit von Mengen die Äquivalenz entspricht:

$N_1 \subseteq N_2$ genau dann, wenn gilt: $x \in N_1 \Rightarrow x \in N_2$.

$N_1 = N_2$ genau dann, wenn gilt: $x \in N_1 \Leftrightarrow x \in N_2$.

Analog zu den aussagenlogischen Verknüpfungen `und' , `oder` gibt es den **Durchschnitt** ($\cap$, **intersect**) und die **Vereinigung** ($\cup$, **union**) von Teilmengen, die aus zwei gegebenen Teilmengen jeweils eine neue Teilmenge machen:

$N_1 \cap N_2 := \{x \mid x \in N_1 \wedge x \in N_2\}$, genauer $N_1 \cap N_2 := \{x \in M \mid x \in N_1 \wedge x \in N_2\}$

$N_1 \cup N_2 := \{x \mid x \in N_1 \vee x \in N_2\}$, genauer $N_1 \cup N_2 := \{x \in M \mid x \in N_1 \vee x \in N_2\}$.

Bei der **Komplementmenge**, die der Negation entspricht, ist der Bezug zur Gesamtmenge schon wichtiger:

$M - N := \mathcal{C}N := \{x \in M \mid x \notin N\}$.

```
> M:={1,4,5,3,7};
  N1:={1,3};
  N2:={7,3};
```

$$M := \{1, 3, 4, 5, 7\}$$

$$N1 := \{1, 3\}$$

$$N2 := \{3, 7\} \quad \textbf{(1.2.1)}$$

```
> N1 intersect N2;
```

$$\{3\} \quad \textbf{(1.2.2)}$$

```
> N1 union N2;
  {op(N1),op(N2)};
```

$$\{1, 3, 7\}$$

$$\{1, 3, 7\} \quad \textbf{(1.2.3)}$$

```
> M minus N1;
```

$$\{4, 5, 7\} \quad \textbf{(1.2.4)}$$

```
> N2 minus N1;
```

$$\{7\} \tag{1.2.5}$$

ÜBUNG [03]:

Die **Differenzmenge** zweier Mengen $N_1, N_2 \subseteq M$ ist definiert als

$$N_1 - N_2 := \{x | x \in N_1 \wedge x \notin N_2\}.$$

1) Wie kann man die Differenzmenge mit Hilfe von $\cap, \mathcal{C}$ (Schnitt, Komplement) darstellen?
2) Finde auch noch eine zweite Darstellung, die mit $\cup$ (Vereinigung) statt $\cap$ arbeitet!
Hinweis: Betrachte zuerst $\mathcal{C}(N_1 - N_2)$
3) Warum ist bei der Bildung der Komplementmenge der Bezug zur Gesamtmenge relevant?
4) Was entspricht dem ausschließenden *oder* auf der mengentheoretischen Seite?

MAPLE: (Schleife über eine endliche Menge) Will man jedem Element einer Menge ein neues Objekt zuweisen und die Menge der resultierenden Objekte betrachten, gibt es einen umständlichen und einen eleganten Weg, dies zu tun:

```
> S:={};
  for i in M do
    S:=S union {{i}}
  end do;
```

$$S := \{\}$$
$$S := \{\{1\}\}$$
$$S := \{\{1\}, \{3\}\}$$
$$S := \{\{1\}, \{3\}, \{4\}\}$$
$$S := \{\{1\}, \{3\}, \{4\}, \{5\}\}$$
$$S := \{\{1\}, \{3\}, \{4\}, \{5\}, \{7\}\} \tag{1.2.6}$$

```
> map(i->{i},M);
```

$$\{\{1\}, \{3\}, \{4\}, \{5\}, \{7\}\} \tag{1.2.7}$$

Sofern möglich wird insbesondere bei größeren Mengen letzterer Weg empfohlen. Du solltest dich generell mit dem Befehl **map** vertraut machen, im Zweifelsfall durch Nachfragen bei den Hilfskräften. Dieser Befehl wird dir in Zukunft häufig begegnen.

```
> map(f, M);
```

$$\{f(1), f(3), f(4), f(5), f(7)\} \tag{1.2.8}$$

```
> ?map
```

ÜBUNG [04]:

1) Verstehe die Idee des **map**-Befehls.

2) Berechne die folgenden Mengen mit Hilfe des **map**-Befehls:
a) Die Menge der ersten 21 positiven geraden Zahlen.
b) Die Menge der ersten 42 positiven Quadratzahlen.
c) Die Menge $\{\{i, i+1\} \mid i \in \{1,.., 42\}\}$.
d) Die Menge $\{1,..., 7\} \times \{1,..., 6\} = \{[i, j] \mid i \in \{1,.., 7\}, j \in \{1,.., 6\}\}$.
Hinweise zu (d): Schlage im Zweifelsfall die Definition des Cartesischen Produktes zweier Mengen im nachfolgenden Abschnitt nach. Benutze zwei ("verschachtelte") **map**s. Wir verwenden für Paare in Maple die Notation $[i, j]$ anstelle von (i, j), welches Maple nicht beherrscht.

MAPLE: Wenn wir die Komplementmenge der einelementigen Teilmengen als Menge generieren wollen:

```
> map(i->M minus {i}, M);
```

$$\{\{1, 3, 4, 5\}, \{1, 3, 4, 7\}, \{1, 3, 5, 7\}, \{1, 4, 5, 7\}, \{3, 4, 5, 7\}\} \qquad \textbf{(1.2.9)}$$

```
> map(i->subsop(i=NULL,M), {$1..nops(M)});
```

$$\{\{1, 3, 4, 5\}, \{1, 3, 4, 7\}, \{1, 3, 5, 7\}, \{1, 4, 5, 7\}, \{3, 4, 5, 7\}\} \qquad \textbf{(1.2.10)}$$

Potenzmengen

MATH: Die **Potenzmenge** $\mathrm{Pot}(M)$ einer Menge M ist die Menge aller Teilmengen von M. Mit anderen Worten: Die Elemente der Potenzmenge $\mathrm{Pot}(M)$ sind die Teilmengen von M. Wir wollen experimentell feststellen, wie viele Elemente die Potenzmenge $\mathrm{Pot}(M)$ in Abhängigkeit von der Anzahl der Elemente von M hat. Hier ist ein Programm **Pot(n)** zur Erzeugung der Potenzmenge von $\{1,..., n\}$:

```
> # Das Programm soll die Potenzmenge von {1,...,n} berechnen
  und ausgeben.
  # Pot ist der Name des Programms,
  # n der Parameter, der als Eingabe an das Programm übergeben
  wird,
  # n::nonnegint überprüft, ob n eine nicht negative ganze
  Zahl (nonnegint = non-negative integer) ist.
  Pot:=proc(n::nonnegint)

    # Hier werden lokale Variablen, genauer Namen dafür
  reserviert. Diese Variablen haben nur
    # innerhalb des Programms eine Bedeutung, d.h. es kann
  außerhalb des Programms Variablen
    # mit gleichem Namen geben, die durch die lokalen
  Variablen nicht beeinflusst werden.
    local P1, P2;

    # Im Falle n=0 enthält die Potenzmenge nur die leere Menge
    if n = 0 then
      return {{}};
```

```
    end if;

    # Wenn eine Menge nicht leer ist, bekommt man die
  Potenzmenge dadurch, dass man
    # zunächst ein Element entfernt und die Potenzmenge P1
  dieser Menge berechnet.
    # Anschließend nimmt man die Elemente von P1, fügt das
  anfangs entfernte Element
    # zu jeder der Teilmengen in P1 wieder hinzu und nennt das
  ganze P2. Die Vereinigung
    # von P1 und P2 ist dann die Potenzmenge.

    # Teilmengen von {1,...,n}, die n nicht enthalten.
    P1 := Pot(n-1);
    # Teilmengen von {1,...,n}, die n enthalten.
    P2 := map(a->a union {n}, P1);
    # Alle Teilmengen von {1,...,n}.
    return P1 union P2;
  end proc:
> Pot(0);
```

$$\{\{\}\} \qquad \textbf{(1.3.1)}$$

```
> Pot(3);
```

$$\{\{\}, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\} \qquad \textbf{(1.3.2)}$$

Die Anzahl der Elemente der Potenzmengen $\mathrm{Pot}(3)$:

```
> nops(Pot(3));
```

$$8 \qquad \textbf{(1.3.3)}$$

Die Anzahl der Elemente der Potenzmengen $\mathrm{Pot}(0)$ bis $\mathrm{Pot}(5)$:

```
> map(nops@Pot, [$0..5]);
  map(i -> nops(Pot(i)), [$0..5]);
```

$$[1, 2, 4, 8, 16, 32]$$

$$[1, 2, 4, 8, 16, 32] \qquad \textbf{(1.3.4)}$$

DENKANSTOSS: Wie erklärt sich der Verdopplungseffekt $[1, 2, 4, 8, \ldots]$ aus dem obigen Programm?

MAPLE u. MATH: Vielleicht stört man sich daran, dass wir nur die Potenzmenge von $\{1,\ldots, n\}$ konstruiert haben. Wie konstruieren wir hieraus die Potenzmenge von $\{a, b, c\}$?

```
> subs([1=a,2=b,3=c],Pot(3));
```

$$\{\{\}, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\} \qquad \textbf{(1.3.5)}$$

ÜBUNG [05]:

Das folgende rekursive Programm zur Aufzählung der k-elementigen Teilmengen von $\{1,\ldots, n\}$ hat einen Fehler.

1) Finde und korrigiere diesen Fehler.
2) Verstehe das Programm und füge erläuternde Kommentare ein.
(Hinweis: Das bedeutet, dass du das Programm im Testat erklären kannst! Mache dir insbesondere klar, wofür die Mengen P und Q im Programm stehen.)
3) Rechne die Menge der 2-elementigen Teilmengen der Menge $\{1, 2, 3\}$ mit der Idee des Programmes von Hand aus.
4) Mache dir klar, dass die Mengen P und Q aus dem Programm immer disjunkt sind (Beweis!), sodass für die Anzahl $a(n, k)$ der k-elementigen Teilmengen von $\{1,\ldots, n\}$ (oder allgemeiner (vgl. unten) einer n-elementigen Menge) die Rekursionsgleichung

$$a(n, k) = a(n-1, k-1) + a(n-1, k)$$

aus dem Pascalschen Dreick gilt.
Bemerkung zu (4): Man hat sogar die selben Anfangswerte, sodass $a(n, k) = \binom{n}{k}$ ist.

```
> Potn:=proc(n::nonnegint,k::nonnegint)
    local P,Q,S;
    if k=0 then
      return {};
    end if;
    if k=n then
      return {{$1..n}};
    end if;
    P:=Potn(n-1, k-1);
    P:=map(S->S union {n}, P);
    Q:=Potn(n-1, k);
    return P union Q;
  end proc:
```

```
> Potn(4,0);
```

$$\{\{\}\} \tag{1.3.6}$$

```
> Potn(7,3);
```

$$\{\{1, 2, 3\}, \{1, 2, 4\}, \{1, 2, 5\}, \{1, 2, 6\}, \{1, 2, 7\}, \{1, 3, 4\}, \{1, 3, 5\}, \{1, 3, 6\}, \{1, 3, 7\}, \{1, 4, 5\}, \{1, 4, 6\}, \{1, 4, 7\}, \{1, 5, 6\}, \{1, 5, 7\}, \{1, 6, 7\}, \{2, 3, 4\}, \{2, 3, 5\}, \{2, 3, 6\}, \{2, 3, 7\}, \{2, 4, 5\}, \{2, 4, 6\}, \{2, 4, 7\}, \{2, 5, 6\}, \{2, 5, 7\}, \{2, 6, 7\}, \{3, 4, 5\}, \{3, 4, 6\}, \{3, 4, 7\}, \{3, 5, 6\}, \{3, 5, 7\}, \{3, 6, 7\}, \{4, 5, 6\}, \{4, 5, 7\}, \{4, 6, 7\}, \{5, 6, 7\}\} \tag{1.3.7}$$

```
> nops(Potn(7,3)) = binomial(7,3);
  # Hinweis: es sollte tatsächlich 35 herauskommen, nicht 15!
```

$$35 = 35 \tag{1.3.8}$$

MATH: Es gehört zum Wesen der Mathematik, dass ihre Begriffsbildungen nicht

isoliert zu sehen sind, sondern eine dichte Vernetzung der Begriffe vorhanden ist, die ein ganzes Gebäude von Begriffen darstellt. Die Beziehungen der Begriffe untereinander sind oft wichtiger als die Begriffe selbst. Um eine Ahnung von diesen Beziehungen verschiedener mathematischer Disziplinen untereinander zu erzeugen, wollen wir an dieser Stelle ein Beispiel angeben, wo Potenzmengen endlicher Mengen (für den Neuling wahrscheinlich überraschend) eine Rolle spielen:

```
> mul((y+x[i]),i=1..3);
> expand(%);
> collect(%,y);
```

$$(y+x_1)\,(y+x_2)\,(y+x_3)$$

$$y^3+y^2\,x_1+y^2\,x_2+y^2\,x_3+y\,x_1\,x_2+y\,x_1\,x_3+y\,x_2\,x_3+x_1\,x_2\,x_3$$

$$y^3+(x_1+x_2+x_3)\,y^2+(x_1\,x_2+x_1\,x_3+x_2\,x_3)\,y+x_1\,x_2\,x_3 \qquad \textbf{(1.3.9)}$$

Man betrachte die Koeffizienten von y^j und denke an $(3-i)$-elementige Teilmengen von $\{1, 2, 3\}$. Dies wird zu einem späteren Zeitpunkt im Praktikum noch vertieft werden.

DENKANSTOSS: Man mache sich die Verhältnisse am Dreieck klar: Warum ist es sinnvoll, ausgehend von der Eckenmenge die Kanten als zweielementige Teilmengen anzusehen? Wie kann man Analoges für vierelementige Mengen machen?

2.) Cartesische Produkte und Abbildungen

Aufgaben: 4

```
> restart;
```

Cartesische Produkte von (endlichen) Mengen

Eine wichtige Art, aus zwei vorgegebenen Mengen eine neue Menge zu bilden, ist das Cartesische Produkt, welches seinerseits Ausgangspunkt für eine Reihe anderer Begriffsbildungen ist, die Vergleiche zwischen Mengen gestatten. Der Vereinigung disjunkter Mengen entspricht bei den natürlichen Zahlen die Addition, der Bildung des Cartesischen Produktes die Multiplikation.

MATH: Fast noch wichtiger als die Mengen selbst sind Vergleiche, die man zwischen den Mengen anstellt. Die genauen Konzepte heißen Relationen und Abbildungen. Abbildungen sind gleichbedeutend mit Funktionen. Sie stellen den wichtigsten Spezialfall von Relationen dar, auf welchen wir gleich noch eingehen werden.

MATH: Sind M und N Mengen, so ist das **Cartesische Produkt** $M \times N$ die Menge aller geordneten Paare (m, n) mit $m \in M$ und $n \in N$. (Geordnet bedeutet $(m, n) = (s, t)$ genau dann, wenn $m = s$ und $n = t$ gilt.)

MAPLE kann die geordneten Paare nicht wie sonst üblich in runde Klammern schreiben:

```
> (1,2);
```

$$1, 2 \tag{2.1.1}$$

Wir benutzen daher Listen mit zwei Einträgen als Notation:

```
> [1,2];
```

$$[1, 2] \tag{2.1.2}$$

```
> evalb([1,2]=[2,1]);
```

$$\mathit{false} \tag{2.1.3}$$

Im Unterschied zu ungeordneten Paaren, die man als Mengen darstellen kann:

```
> evalb({1,2}={2,1});
```

$$\mathit{true} \tag{2.1.4}$$

Hier ist eine Prozedur, die das Cartesische Produkt zweier endlicher Mengen konstruiert:

```
> CartProd:=proc(A,B)
    local x,y,CP;
    CP:=NULL;
    for x in A do
      for y in B do
        CP:=CP,[x,y];
      end do
    end do;
    return {CP}
  end proc:
```

Es geht auch deutlich kürzer mit **map** (siehe Übung [04] im Abschnitt "Grundbegriffe: Elemente und Teilmengen").

```
> X:={1,2};
  Y:={a,b,c};
```

$$X := \{1, 2\}$$
$$Y := \{a, b, c\} \tag{2.1.5}$$

```
> CartProd(X,Y);
  CartProd(Y,X);
```

$$\{[1, a], [1, b], [1, c], [2, a], [2, b], [2, c]\}$$
$$\{[a, 1], [a, 2], [b, 1], [b, 2], [c, 1], [c, 2]\} \tag{2.1.6}$$

```
> nops(%%),nops(%);
```

$$6, 6 \tag{2.1.7}$$

VISUAL: Es ist besser, sich die Elemente von $X \times Y$ in einem rechteckigen Schema (Matrix) angeordnet vorzustellen, dessen Zeilen durch X und dessen Spalten durch Y festgelegt sind:

```
> Matrix(nops(X),nops(Y),(i,j)->``(X[i],Y[j]));
```

$$\begin{bmatrix} (1, a) & (1, b) & (1, c) \\ (2, a) & (2, b) & (2, c) \end{bmatrix} \quad \textbf{(2.1.8)}$$

Man beachte: Dies ist nur eine Visualisierung. Die mathematische Struktur, mit der man arbeitet, ist in der Prozedur **CartProd** definiert. Die Visualisierung stellt zum Beispiel sofort in Evidenz, dass $|X \times Y| = |X| \cdot |Y|$ ist, d. h.

```
> evalb(nops(X)*nops(Y)=nops(CartProd(X,Y)));
```

true **(2.1.9)**

, wobei |M| die Anzahl der Elemente der Menge M sei. Wir werden bald einsehen, dass die Anzahl der Elemente einer Menge wohldefiniert ist.

ÜBUNG [01]:

Seien X_1, X_2 Teilmengen einer Menge X und Y_1, Y_2 Teilmengen einer Menge Y.
Beweise oder widerlege:
1) $(X_1 \times Y_1) \cap (X_2 \times Y_2) = (X_1 \cap X_2) \times (Y_1 \cap Y_2)$,
2) $(X_1 \times Y_1) \cup (X_2 \times Y_2) = (X_1 \cup X_2) \times (Y_1 \cup Y_2)$,
wobei diese beiden Mengen und auch ihr Schnitt als Teilmengen von $X \times Y$ zu sehen sind.
Hinweise: Eine Skizze könnte hilfreich sein. Beweise werden immer allgemein geführt, Gegenbeispiele sollten aber so leicht wie möglich gehalten werden.

Abbildungen (oder Funktionen) als besondere Relationen

MATH: Teilmengen des Cartesischen Produktes $X \times Y$ von X und Y nennt man oft auch **Relationen** zwischen X und Y.

In dieser Allgemeinheit sind Relationen nicht so wichtig - sie werden es erst, wenn man noch weiter spezialisiert. Wir kommen damit zu dem zentralen, wichtigen und grundlegenden mathematischen Konzept der **Funktion** oder der **Abbildung**.

Die Abbildungen, die du aus der Schule kennst, lassen sich meistens durch geschlossene Ausdrücke beschreiben. Was wir hier brauchen ist sehr viel allgemeiner!

MATH: Eine **Abbildung** oder **Funktion** f der Menge X in die Menge Y, in Zeichen:
$f: X \to Y$,
ist eine Relation zwischen X und Y, also eine Teilmenge von $X \times Y$ mit der Eigenschaft:
Für jedes $x \in X$ gibt es genau ein $y \in Y$ mit $(x, y) \in f$.

Mit Quantoren:
$\forall\ x \in X\ \exists$ genau ein $y \in Y$ mit $(x, y) \in f$.
Man schreibt dafür auch:
$\forall\ x \in X \exists! y \in Y$ mit $(x, y) \in f$.
Wir schreiben dann auch $y = f(x)$, weil dieses y ja eindeutig durch x festgelegt ist. Dabei heißt X der **Definitionsbereich** und Y der **Wertebereich** von f.

In der Sprache der Matrix, die wir oben gesehen haben, heißt das, dass die Teilmenge von $X \times Y$ so geartet ist, dass in jeder Zeile nur genau ein Eintrag zu f gehört. "Genau einer'' heißt in diesem Kontext immer zweierlei: "höchstens einer'' und "mindestens einer''. Wenn wir uns $X \times Y$ wieder als Matrix vorstellen, wobei die Zeilen durch X und die Spalten durch Y festgelegt sind, so heißt dies, dass wir solche Teilmengen von $X \times Y$ betrachten, die in jeder Zeile genau ein Element haben. Hier ist ein Beispiel:

```
> X:={$1..3};
  Y:={a,b};
```

$$X := \{1, 2, 3\}$$
$$Y := \{a, b\} \qquad \textbf{(2.2.1)}$$

```
> f:={[1,a],[2,a],[3,b]};
```

$$f := \{[1, a], [2, a], [3, b]\} \qquad \textbf{(2.2.2)}$$

MAPLE: f ist offensichtlich eine Abbildung von X nach Y. Es ist aber völlig klar, dass diese Datenstruktur zu umständlich ist, um effektiv damit zu arbeiten. Selbst die Bestimmung eines Funktionswertes ist schon sehr umständlich:

```
> Wert:=proc(F::set,x)
    local y,z;
    for y in F do
      if y[1]=x then
        return y[2];
      end if;
    end do;
  end proc:
> Wert(f,1);
```

$$a \qquad \textbf{(2.2.3)}$$

MATH: Die programmtechnische Umständlichkeit tut der grundlagenmathematischen Nützlichkeit der Abbildungsdefinition keinen Abbruch. Wir wollen diese Abbildung visualisieren - zunächst das gesamte Cartesische Produkt, dann die Abbildung:

```
> Matrix(nops(X),nops(Y),(i,j)->[X[i],Y[j]]);
```

$$\begin{bmatrix} [1, a] & [1, b] \\ [2, a] & [2, b] \\ [3, a] & [3, b] \end{bmatrix} \qquad \textbf{(2.2.4)}$$

```
> Matrix(nops(X),nops(Y),(i,j)->if [X[i],Y[j]] in f then [X
```

```
  [i],Y[j]] else 0 end if);
```

$$\begin{bmatrix} [1, a] & 0 \\ [2, a] & 0 \\ 0 & [3, b] \end{bmatrix} \quad \textbf{(2.2.5)}$$

MATH: Wir sehen in jeder Zeile genau einen Eintrag ungleich 0. Dies ist mit der Definition des Begriffes Abbildung zu vergleichen. Hätten wir die Zeilen durch die Elemente von X und die Spalten durch die Elemente von Y indiziert, so hätte es genügt, ein Kreuz an den Stellen zu machen, wo etwas ungleich Null steht: Die gesamte Information wäre vorhanden gewesen. Hier nochmals eine andere Visualisierung, wo Zeilen und Spalten vertauscht sind.

```
> LinearAlgebra[Transpose](%);
```

$$\begin{bmatrix} [1, a] & [2, a] & 0 \\ 0 & 0 & [3, b] \end{bmatrix} \quad \textbf{(2.2.6)}$$

MATH: Dies kommt der üblichen Visualisierung des Funktionsgraphen sehr nahe, der genau das ist, was wir als Funktion oder Abbildung definiert haben:

```
> printf("          ---|--------------|\n");
  printf("           a | x    x       |\n");
  printf("           b |           x  |\n");
  printf("          ---|--------------|\n");
  printf("             | 1    2    3  |\n");
```

```
          ---|--------------|
           a | x    x       |
           b |           x  |
          ---|--------------|
             | 1    2    3  |
```

ÜBUNG [02]:

1) Gib aus dieser Tabelle heraus den Definitionsbereich und Wertebereich von f an.
2) Woran erkennt man jetzt, dass f eine Abbildung ist?

MATH: Übrigens lässt sich an dieser Beschreibung von Abbildungen auch leicht zählen, wie viele Abbildungen es von X nach Y gibt: Es gibt $|X|$ Spalten in unserer Tabelle. Die Regel besagt: In jeder Spalte müssen wir genau ein Kreuz machen. Das gibt $|Y|$ Möglichkeiten für jede Spalte. Also haben wir insgesamt $|Y|^{|X|}$ Möglichkeiten. Übrigens ist dies der Grund, warum man die Menge der Abbildungen von X nach Y mit Y^X bezeichnet. Es gilt dann also $|Y^X| = |Y|^{|X|}$.

PROJEKT (für die Freizeit): Übertrage den gerade geführten Beweis in ein Programm, welches alle Abbildungen von X nach Y konstruiert.

Darstellungsmöglichkeiten von Abbildungen

MAPLE hat die Möglichkeit, Funktionen auch anders darzustellen, eben als Zuordnung. Die Spezifikation des Definitions- und Wertebereiches bleibt dabei manchmal offen:

```
> g:=x->x^2;
```

$$g := x \to x^2 \qquad \textbf{(2.3.1)}$$

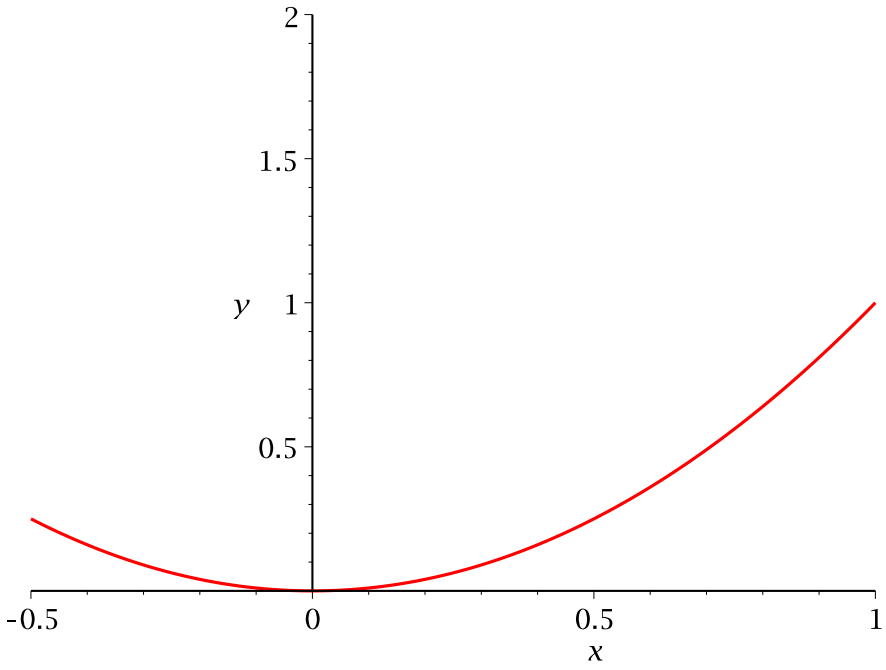
```
> g(1);
```

$$1 \qquad \textbf{(2.3.2)}$$

```
> g(0);
```

$$0 \qquad \textbf{(2.3.3)}$$

```
> plot(g(x),x=-.5..1,y=0..2);
```



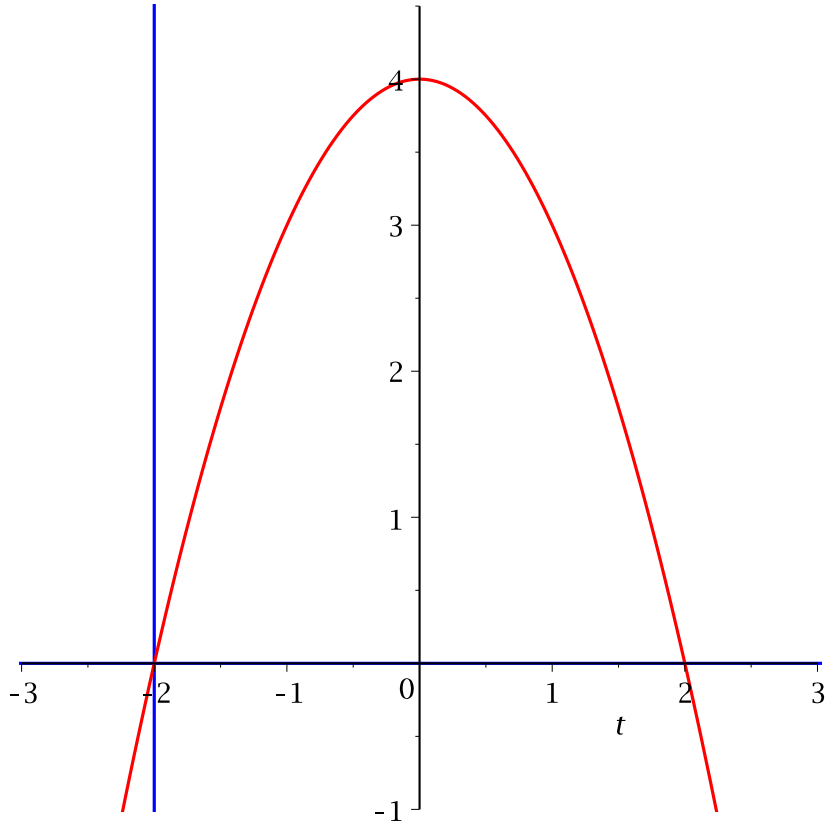
Dies hat man so zu interpretieren: g ist eine Abbildung des abgeschlossenen Intervalles $\left[-\frac{1}{2}, 1\right]$ in das abgeschlossene Intervall $[0, 2]$. Das Cartesische Produkt der Intervalle ist als Rechteck dargestellt und die Funktion g als Teilmenge (in rot) davon: die Menge der Punkte $(x, g(x))$, wobei x den Definitionsbereich durchläuft.

MATH u. MAPLE: Die folgende Animation demonstriert nochmals den Zusammenhang zwischen dem mehr statischen Standpunkt, dass eine Funktion eine Relation ist und dem mehr dynamischen Zuordnungsstandpunkt:
(Halte dich nicht daran auf, das Programm zu analysieren, welches die Animation erstellt.)

```
> parab:=proc(r)
    local S, P;
    S := plots[animate]([t,x,x=-r^2-1..r^2+1],t=-abs(r)..abs
  (r),numpoints=100,frames=100,color=blue):
    P := plots[animate]([x,4-t^2,x=-r^2-1..r^2+1],t=-abs(r)..
  abs(r),numpoints=100,frames=100, color=blue):
    plots[display]({S,P,plot(4-t^2,t=-abs(r)-1..abs(r)+1,-1..
  r^2+0.5)});
  end proc:
```

Klicken nach Erstellung des Bildes mit der linken Maustaste in das Bild und dann oben auf die Taste "Play the animation" (Gelbes Dreieck / Play-Taste).

```
> parab(2);
```

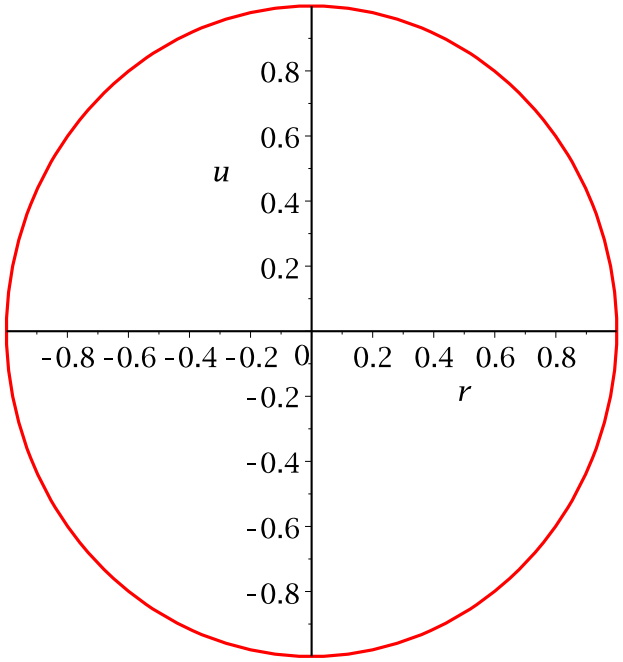


Der Definitionsbereich liegt auf der waagerechten Achse, der Wertebereich auf

der senkrechten. Die Schnittpunkte der beiden beweglichen Geraden mit den Achsen geben den aktuellen Urbildpunkt im Definitionsbereich und den zugehörigen Bildpunkt im Bildbereich an. Für physikalisch Interessierte: Interpretiert man den Definitionsbereich als Zeit (mit etwas sonderbar gewählter Anfangszeit), so sieht man im Wertebereich eine Simulation eines Wurfes.

Die folgende Relation ist keine Abbildung von $[-1, 1]$ nach $[-1, 1]$:

```
> plots[implicitplot](r^2+u^2=1,r=-1..1,u=-1..1);
```



ÜBUNG [03]:

1) Warum ist diese Relation keine Abbildung?
2) Mache aus dem letzten Beispiel zwei Abbildungen, deren Vereinigung gerade die obige Relation ergibt.
3) Plotte beide Funktionen und mache dabei anschaulich klar, dass die Vereinigung der Funktionen die obige Relation ergibt.
Hinweis: Die beiden Funktionen enthalten Wurzeln. Beachte, dass für nichtnegatives x in Maple $\sqrt{x}$ wieder als nichtnegative Zahl definiert ist.

MAPLE u. MATH: Ein wichtiger Typ von Abbildungen sind (endliche) **Folgen**, genauer Y-wertige Folgen der Länge n.

Dies sind Abbildungen a von $\{1,\ldots, n\}$ in die Menge Y, also
$a: \{1, 2,\ldots, n\} \to Y$.
(Wir haben Folgen bereits etwas weniger formell im Abschnitt "Grundbegriffe: Elemente und Teilmengen" behandelt.)

In der Mathematik bezeichnet man meistens den Funktionswert an der Stelle i nicht mit $a(i)$ (was völlig legal, aber unüblich, ist), sondern mit a_i.
Die angemessene Datenstruktur in Maple ist die Liste, also eine Maple-Folge, die man in eckige Klammern schreibt: Statt
$\{(i, a_i) \mid i \in \{1, 2,\ldots, n\}\}$
schreibt man
$[a_1, a_2,\ldots, a_n]$
und markiert das i durch die Stelle, an welcher der Folgenwert steht. Dadurch wird es möglich, das i-te Folgenglied, also den Funktionswert an der Stelle i, schnell abzurufen:

```
> f:=[1,2,1,2,3,5];
```

$$f := [1, 2, 1, 2, 3, 5] \quad \textbf{(2.3.4)}$$

```
> f[3];
```

$$1 \quad \textbf{(2.3.5)}$$

MAPLE: Eine Umformung in die alte Datenstruktur als Teilmenge des Cartesischen Produktes geschieht so:

```
> map(i->[i,f[i]],{$1..nops(f)});
```

$$\{[1, 1], [2, 2], [3, 1], [4, 2], [5, 3], [6, 5]\} \quad \textbf{(2.3.6)}$$

MAPLE: Der Übergang der Zuordnungschreibweise von oben, wo wir uns stillschweigend $\{1, 2,\ldots, n\}$ als Definitionsbereich vorstellen, zu der Listenschreibweise ist sehr einfach:

```
> eval(g);
```

$$x \to x^2 \quad \textbf{(2.3.7)}$$

```
> map(g,[$1..6]);
```

$$[1, 4, 9, 16, 25, 36] \quad \textbf{(2.3.8)}$$

Zwei sehr wichtige Funktionen: Identität und charakteristische Funktion

MATH: Eine sehr wichtige Abbildung ist die **Identitätsabbildung** einer Menge X. Sie hat X sowohl als Definitions- als auch als Wertebereich und bildet jedes Element auf sich selbst ab.

```
> Id:=x->x;
```

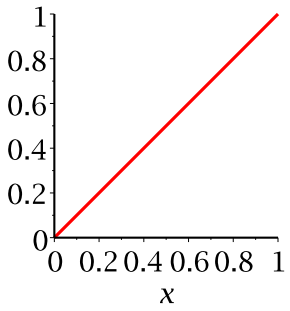
$$Id := x \to x \quad \textbf{(2.4.1)}$$

```
> Id(42);
```

(2.4.2)

(2.4.2)

```
> plot(Id(x),x=0..1,scaling=constrained);
```



MATH: Abbildungen kann man auch zur Beschreibung von Teilmengen heranziehen. Der Teilmenge T der Menge X ordnet man ihre **charakteristische Funktion** $\chi_T: X \to \{0, 1\}$ zu, definiert durch $(T \times \{1\}) \cup ((X - T) \times \{0\})$:

```
> X:={$1..10};
  T:={$3..7};
```

$$X := \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$$

$$T := \{3, 4, 5, 6, 7\} \quad \textbf{(2.4.3)}$$

```
> chi:=i->if i in T then 1 else 0 end if;
```

$$\chi := i \to \textbf{if } i \in T \textbf{ then } 1 \textbf{ else } 0 \textbf{ end if} \quad \textbf{(2.4.4)}$$

```
> chi(3), chi(2);
```

$$1, 0 \quad \textbf{(2.4.5)}$$

```
> map(chi,[$1..10]);
```

$$[0, 0, 1, 1, 1, 1, 1, 0, 0, 0] \quad \textbf{(2.4.6)}$$

DENKANSTOSS: Offensichtlich kann man aus der charakteristischen Funktion die Teilmenge rekonstruieren. Wie würde man in der Sprache der charakteristischen Funktionen Durchschnitte, Vereinigungen und Komplementbildung ausdrücken?

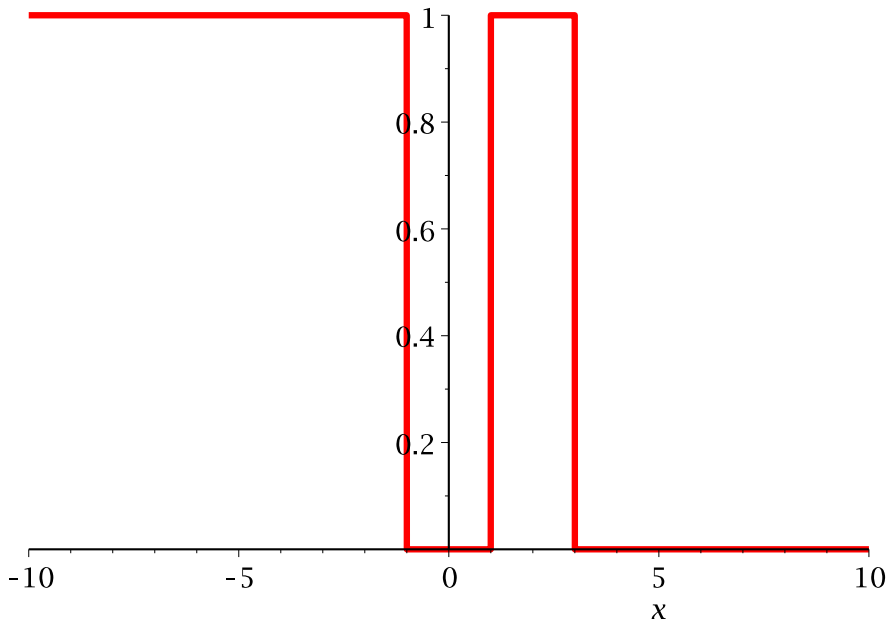
DENKANSTOSS: Benutze die charakteristische Funktion, um zu beweisen, dass die Potenzmenge von $\{1,\ldots, n\}$ genau 2^n Elemente hat.

MAPLE: Bei Teilmengen der reellen Zahlengeraden, die Vereinigungen von Intervallen sind, bietet sich eine besonders einfache Möglichkeit, die charakteristische Funktion zu konstruieren:

```
> f:=x->piecewise(x<-1,1,x<1,0,x<3,1,0):
> 'f(x)'=f(x);
```

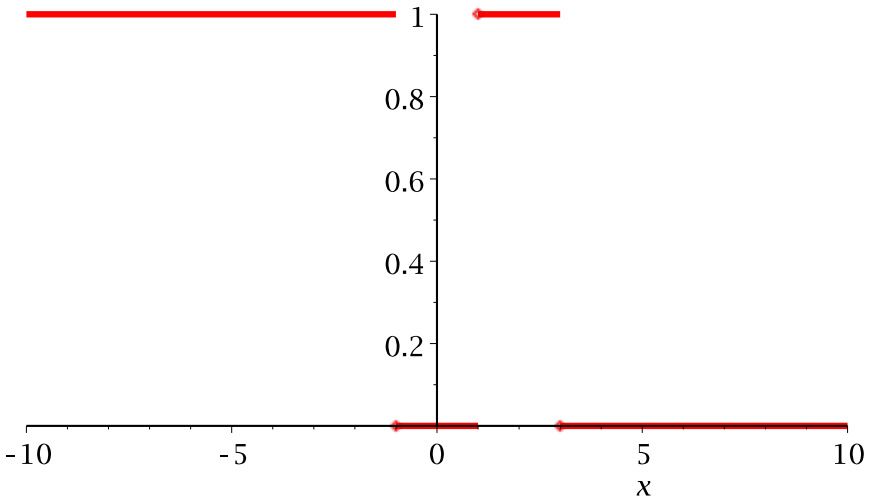
$$f(x) = \begin{cases} 1 & x < -1 \\ 0 & x < 1 \\ 1 & x < 3 \\ 0 & otherwise \end{cases} \qquad \textbf{(2.4.7)}$$

```
> plot(f(x),x=-10..10,thickness=3);
```



Zu der Visualisierung ist kritisch anzumerken, dass die senkrechten roten Striche nicht zur Funktion gehören. Durch eine Zusatzoption kann man Maple veranlassen, erst die Unstetigkeitsstellen der Funktion zu bestimmen, um dort die notwendigen Sprünge zu realisieren:

```
> plot(f(x),x=-10..10,thickness=3,discont=true);
```



Einschränkungen von Abbildungen auf Teilmengen

MATH: Ist $f := X \to Y$ eine Abbildung und $T \subseteq X$ eine Teilmenge von X, so heißt

$$f_{|T} := f \cap (T \times Y)$$

die **Einschränkung** von f auf T. Es ist klar: $f_{|T}$ ist eine Abbildung von T nach Y, in Symbolen:: $f_{|T}: T \to Y$.

MAPLE: Die Einschränkung von Folgen der Länge n auf Folgen der Länge $k < n$ ist in der Listenschreibweise besonders leicht zu realisieren:

```
> f:=[seq(i^3,i=1..7)];
```

$$f := [1, 8, 27, 64, 125, 216, 343] \quad \textbf{(2.5.1)}$$

```
> f[1..3];
```

$$[1, 8, 27] \quad \textbf{(2.5.2)}$$

```
> f[[1,2,5,6]];
```

$$[1, 8, 125, 216] \quad \textbf{(2.5.3)}$$

MAPLE: Da wir ansonsten den Definitionsbereich meistens nicht explizit deklarieren, merkt man oft gar nicht, ob man von einer Einschränkung einer Abbildung oder der ursprünglichen Abbildung spricht. Man muss dann wissen, was man tut und darf den Überblick nicht verlieren.

```
> f:=x->x^2;
```

$$f := x \to x^2 \quad \textbf{(2.5.4)}$$

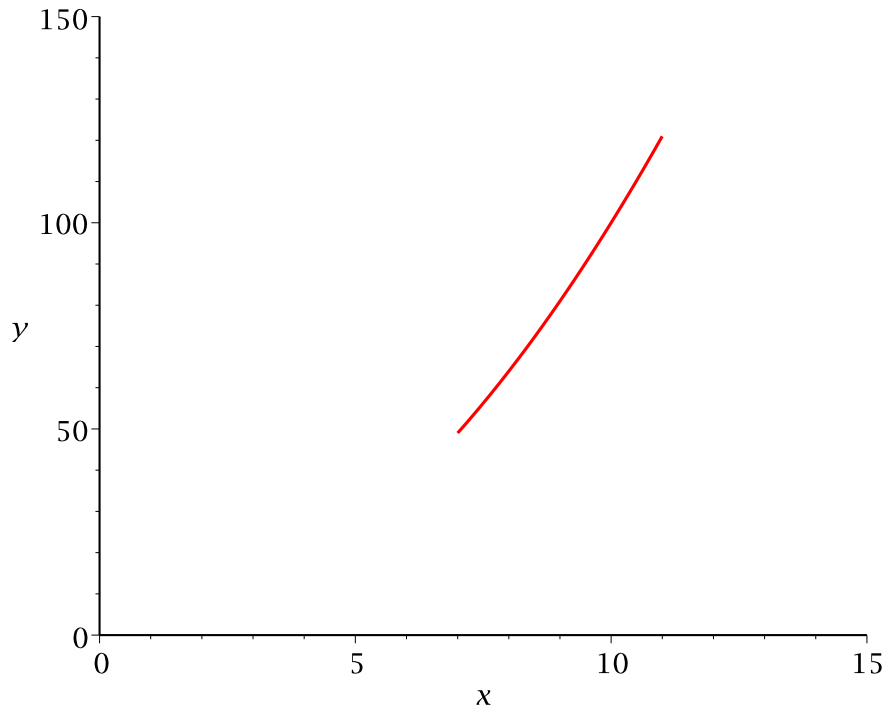
So nimmt *f* beispielsweise als Abbildung der Menge aller reellen Zahlen in sich ein ganz anderes Minimum an als seine Einschränkung auf das abgeschlossene Intervall $[7, 11]$, wohingegen die Einschränkung auf das offene Invervall $(7, 11)$

kein Minimum annimmt, wie ihr in der Analysisvorlesung sehen werdet.
In MAPLE kann man die Einschränkung von f auf $[7, 11]$ wie folgt darstellen:

```
> fI:=piecewise(7<=x and x<=11,f(x),undefined);
```

$$fI := \begin{cases} x^2 & 7 \le x \text{ and } x \le 11 \\ undefined & otherwise \end{cases} \qquad \textbf{(2.5.5)}$$

```
> plot(fI,x=0..15,y=0..150);
```



ÜBUNG [04]:

Sei $f: \{1, 2, 3, 4, 5, 6, 7\} \to \mathbb{Z} : x \mapsto -x^3 - x^2 + 69 \cdot x - 153$ und weiter sei
$T := \{a \in \{1, 2, 3, 4, 5, 6, 7\} \mid a \text{ ist Primzahl}\}$.
Bestimme jeweils den maximalen Funktionwert, den f und $f_{|T}$ annehmen.