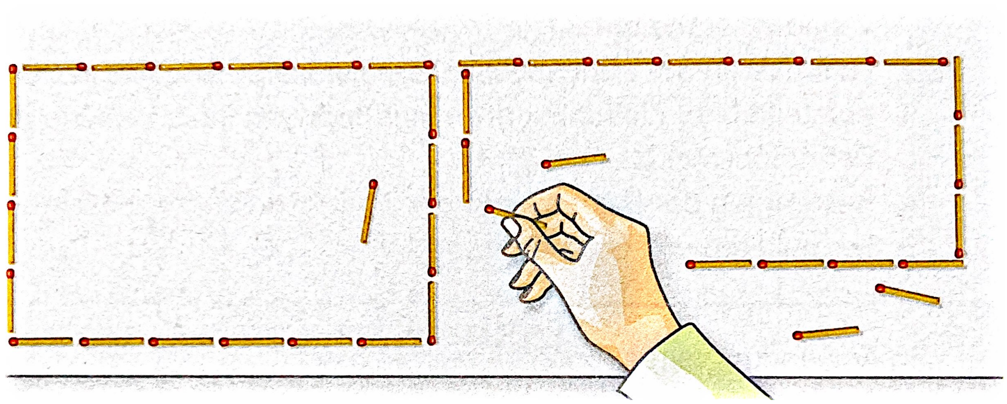
6. Extremwertprobleme mit Nebenbedingungen



Aufgabe:

Aus 20 Streichhölzern soll ein Rechteck mit größtmöglichen Flächeninahlt gelegt werden.

Lösungsvorschläge:

Lösung:

1. Term, von welchem das Maximum bestimmt weden soll: $A(a, b) = a \cdot b$. (a ist die Länge und b die Breite des Rechtecks.)

$$A(a, b) = a \cdot b$$

2. Die Länge a und die Breite b hängen voneinander ab.

$$2 \cdot a + 2 \cdot b = 20$$

3. Umformen der Zusammenhangsgleichung nach einer Variable.

$$a = 10 - b$$

4. Durch Einsezten der umgeformten Zusammenhangsgleichung in die zu maximierende Gleichung wird eine Variable eleminiert.

$$A(b) = (10 - b) \cdot b = 10b - b^2$$

Die Zielwertfunktion entsteht.

5. Extremwertuntersuchung der Zielwertfunktion.

$$A'(b) = 10 - 2b$$

Notwendige Bedingung:

$$10 - 2b = 0$$

$$b = 5$$

Hinreichende Bedingung:

$$A''(b) = -2$$

$$A''(5) = -2$$

Maximalstelle bei b = 5. Der Maximalwert des Flächeninhalts beträgt bei 5 Steichhölzern für die Breite

$$A(5) = 25$$

6. An den Rändern b=0 oder b = 10 könnte ein größeres Maximum vorhanden sein. Überprüfen:

$$A(0) = 0$$

$$A(10) = 0$$

7. Ergebnis lautet: Bei einer Breite von 5 Streichhölzern und einer Höhe von 5 Streichhölzern wird der Flächeninhalt des gelegten Rechtecks maximal.

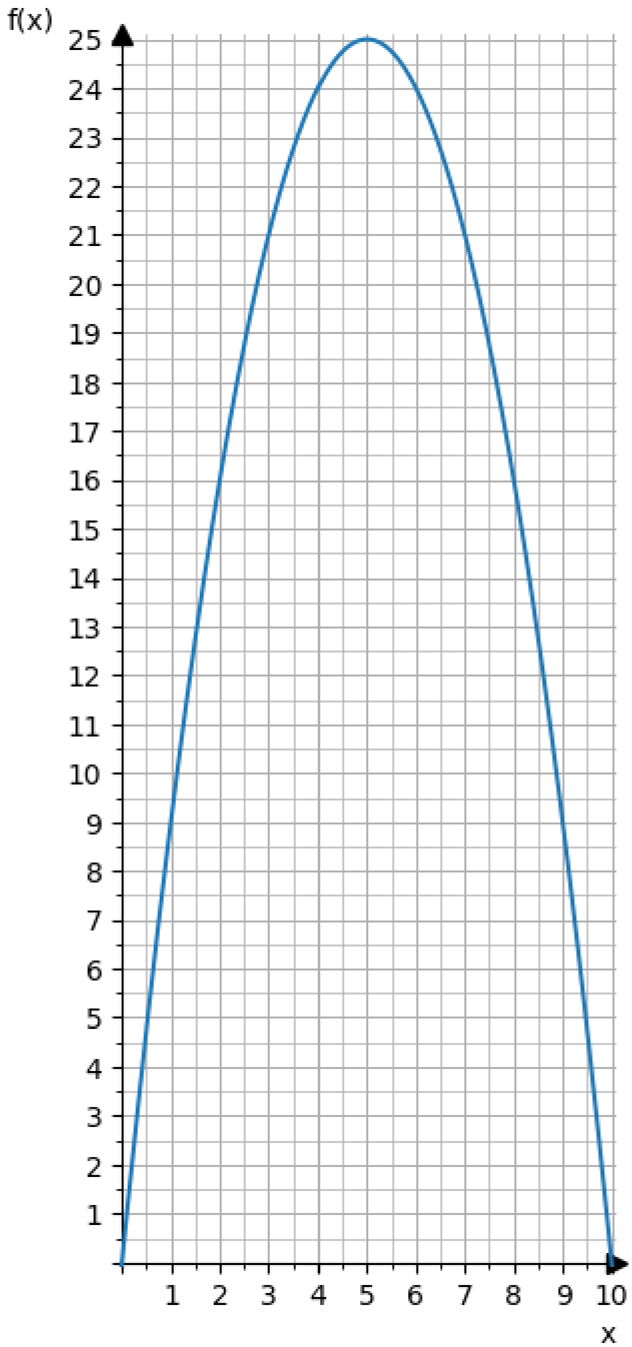
In [3]:

```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.ticker import AutoMinorLocator, MultipleLocator, FuncFormatt

# Defintionsmenge und Funktion
# ----------------------------
a= -0.1 # untere x-Intervallgrenze
b= 10.1 # obere x-Intervallgrenze
c = -0.1# untere y-Intervallgrenze
d = 25.1 # obere y-Intervallgrenze
x = np.linspace(a, b,1000)
y1= -x**2+10*x
# ----------------------------

# Einstellung des Graphen
fig=plt.figure(figsize=(8,8))
ax = fig.add_subplot(1,1,1, aspect =1)

# Definiton der Haupteinheiten, reele Zahlen ohne die 0
def major_tick(x, pos):
    if x==0:
        return ""
    return int(x)

# Achsenskalierung
ax.xaxis.set_major_locator(MultipleLocator(1))
ax.xaxis.set_minor_locator(AutoMinorLocator(2))
ax.yaxis.set_major_locator(MultipleLocator(1))
ax.yaxis.set_minor_locator(AutoMinorLocator(2))
ax.xaxis.set_major_formatter(FuncFormatter(major_tick))
```

```python
ax.yaxis.set_major_formatter(FuncFormatter(major_tick))

# Position der Achsen im Schaubild
ax.spines[['top','right']].set_visible(False)
ax.spines[['bottom','left']].set_position('zero')

# Pfeile für die Achsen
ax.plot((1),(0), ls="", marker= ">", ms=7, color="k", transform=ax.get_yaxis
ax.plot((0),(1), ls="", marker= "^", ms=7, color="k", transform=ax.get_xaxis

# Achsenlänge und Beschriftung
ax.set_xlim(a,b)
ax.set_ylim(c, d)
ax.set_xlabel("x", loc="right")
ax.set_ylabel("f(x)", loc="top", rotation=0)

# Kästchen
ax.grid(linestyle="-", which="major",linewidth=0.7, zorder=-10)
ax.grid(linestyle="-", which="minor",linewidth=0.5, zorder=-10)

# Plot der Funktion
ax.plot(x,y1, zorder=10)
#plt.show()
```

Out[3]:

```
[<matplotlib.lines.Line2D at 0x124942a10>]
```



Definition:

Gegeben ist eine Funktion f auf einem Intervall $I = [a; b]$. Die Randwerte $f(a)$ und $f(b)$ sind immer lokale Extremwerte von f und werden als **Randextrema** bezeichnet.

Strategie zur Lösung von Extremwertproblemen mit Nebenbedingung

1. **Aufstellen des Term, der maximiert werden soll.** Dieser Term kann mehrere Variablen enthalten.
2. **Formulierung der Nebenbedinung.** Diese Beschreibt die Abhängigkeit der beiden Variablen.

3. **Aufstellen der Zielfunktion** wobei eine Variable wegfällt. Die Definitionsmenge muss angegeben werden.
4. **Bestimmung der Extremwerte der Zielfunktion unter Beachtung der Randwerte**
5. **Formulieren des Ergebnisses.**