In [ ]:

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import minimize
```

Analytic center

A LP can have many optimal solutions. For example, the set of points on a playing die that lies furthest from the top of a table is a facet (and not a single vertex).

A combinatorial approach (simplex algorithm) for finding an optimum will terminate in a vertex solution. The interior point method converges to the analytic center of the set of optimal solutions. Let us give the proper definition.

Let

$$Ax \leq b$$

be a system of n linear inequalities, and let $\Phi = \{x; Ax \leq b\}$ be the set of its feasible solutions. We denote $s(x) = b - Ax$. Let $I \subseteq \{1, \ldots, n\}$ be the set of coordinates/indices, for which there exists $x \in \Phi$, so that $(Ax)_i < b_i$ or equivalently $s(x)_i > 0$.

The unique vector $x \in \Phi$ which maximizes

$$\prod_{i \in I} s(x)_i$$

is called the analytic center of a system of linear inequalities (1).

3. Show that there exists $x \in \Phi$ so that for all $i \in I$ we have $s(x)_i > 0$.

For $Ax < b$ we know that the solution lies in the interior of feasibility set. We have to show that there exists $x \in \Phi$ so that for all $i \in I$ the solution lies in the interior.

We can use the fact that the set of feasible solutions Φ is a polytope, which means that it is a bounded and convex set defined by a finite number of linear inequalities. Since it is convex we can take any point:

$$x = \sum_{i \in I} \alpha_i x_i$$

such that $\sum_{i \in I} \alpha_i = 1, \forall \alpha_i > 0$. We obtain a point x that lies in the interior of our feasible set and for all $i \in I$ we have $s(x)_i > 0$.

4. Show that the analytic center optimization problem is equivalent to a strictly convex optimization problem.

Taking the logarithm of the objective function, we can rewrite the optimization problem as:

$$\max_{x \in \Phi} \sum_{i \in I} \log(s(x)_i)$$

Since the logarithm is a strictly increasing function, this transformation does not change the optimal solution. The objective function is now a sum of logarithms, which is a strictly concave function. Since we are maximizing a concave function, this is equivalent to minimizing with a minus sign:

$$\min_{x \in \Phi} - \sum_{i \in I} \log(s(x)_i)$$

The negation of a strictly concave function is a strictly convex function, so the analytic center optimization problem is equivalent to a strictly convex optimization problem.

5. Show that the analytic center is unique.

We can use the fact that we've proven in subtask 2. Since the analytic center optimization problem is equivalent to a strictly convex optimization problem, we know that it has only one global minimum, therefore the analytic center is unique. In our case the original optimization problem has a unique global maximum, since it maximizes a concave function.

6. Find the analytic center for the following system of linear inequalities:

$$\begin{aligned} 2x_1 + 2x_2 &\leq 12 \\ 3x_1 + 1x_2 &\leq 15 \\ x_1 &\geq 0 \\ x_2 &\geq 0 \end{aligned}$$

7. Let us add an additional constraint $2x_1 + 2x_2 \leq 15$ to the above system of inequalities. Clearly this constraint does not change the set of feasible solutions. What happens with the analytic center?

8. Let us add an additional constraint $2x_1 + 2x_2 \leq 12$ to the above system - this one suspiciously resembles one of the original constraints. What happens with the analytic center in this case?

In [ ]:

```
def f1(x):
    x1, x2 = x
    return -(12 - 2*x1 - 2*x2)*(15 - 3*x1 - x2) * x1 * x2

def f2(x):
    x1, x2 = x
    return -(12 - 2*x1 - 2*x2)*(15 - 3*x1 - x2) * x1 * x2 * (15 - 2*x1 - 2*x2)

def f3(x):
    x1, x2 = x
    return -(12 - 2*x1 - 2*x2)*(15 - 3*x1 - x2) * x1 * x2 * (12 - 2*x1 - 2*x2)

def constraint1(x):
    x1, x2 = x
    return 12 - 2*x1 - 2*x2

def constraint2(x):
    x1, x2 = x
```

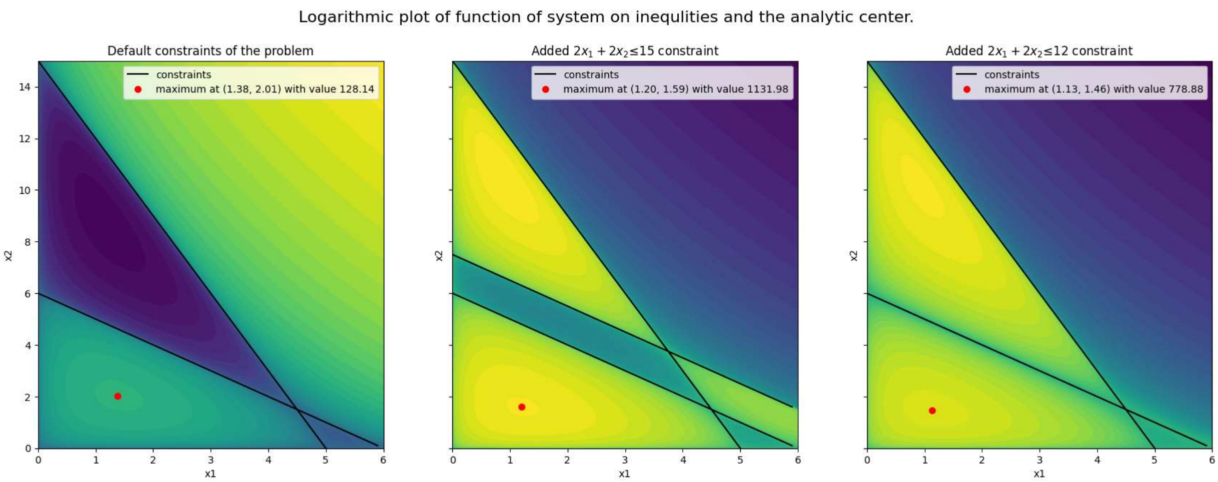
```
    return 15 - 3*x1 - x2

def get_max(f, bounds, constraints, x0=[1,1]):
    return minimize(f, x0, method='SLSQP', bounds=bounds, constraints=constraints).x
```

In [ ]:

```
fig, ax = plt.subplots(1, 3, sharey=True, figsize=(21,7))

x, y = np.linspace(0, 6, 500), np.linspace(0, 15, 500)
X, Y = np.meshgrid(x, y)

x = np.arange(0, 6, 0.1)

y1 = 6 - x
y2 = 15 - 3*x
y3 = (15 - 2*x)/2

for i, f in enumerate([f1, f2, f3]):
    Z = -f([X, Y])
    mask = Z != 0
    Z[mask] = np.sign(Z[mask]) * np.log(np.abs(Z[mask]) + 1)

    ax[i].contourf(X,Y,Z, 50, cmap='viridis')
    ax[i].set_ylim(0,15)

    ax[i].plot(x, y1, '-k', label='constraints')
    ax[i].plot(x, y2, '-k')

    maxima = get_max(f, [(0, 6), (0, 15)], [{'type': 'ineq', 'fun': constraint1}, {'
    ax[i].plot(maxima[0], maxima[1], 'o', color='red', label = 'maximum at ({:.2f},
    ax[i].set_xlabel('x1')
    ax[i].set_ylabel('x2')
    legend = ax[i].legend()

ax[1].plot(x, y3, '-k')

fig.suptitle('Logarithmic plot of function of system on inequlities and the analytic
ax[0].set_title('Default constraints of the problem')
ax[1].set_title('Added $2x_1 + 2x_2 ≤ 15$ constraint')
ax[2].set_title('Added $2x_1 + 2x_2 ≤ 12$ constraint')
```

Out[ ]:

```
Text(0.5, 1.0, 'Added $2x_1 + 2x_2 ≤ 12$ constraint')
```



We show our results with plots. We plot contours of logaritmic (for nicer contours) maximization function. We observe that with adding a constraint that does not change the set of feasible solutions still changes the analytic center. When we add a new constraint to the system, we are adding a new term to the product that defines the analytic center. This new term can change the value of the product even if it does not change the set of feasible solutions.