Discrete Mathematics in Computer Science D2. Advanced Methods for Recurrences

Malte Helmert, Gabriele Röger

University of Basel

Malte Helmert, Gabriele Röger (University of Discrete Mathematics in Computer Science

D2. Advanced Methods for Recurrences

Fibonacci Series – Generating Functions

1 / 30

D2.1 Fibonacci Series – Generating Functions

Discrete Mathematics in Computer Science — D2. Advanced Methods for Recurrences

D2.1 Fibonacci Series – Generating Functions

D2.2 Master Theorem for Divide-and-Conquer Recurrences

Malte Helmert, Gabriele Röger (University of Discrete Mathematics in Computer Science

D2. Advanced Methods for Recurrences Revisiting the Fibonacci Series

- In this section we study generating functions, a powerful method for solving recurrences.
- Generating functions allow us to directly derive closed-form expressions for recurrences.
- Full mastery of generating functions requires solid knowledge of calculus, in particular power series.
- Rather than develop the topic in its full depth, we will look at it within the context of a case study, proving the closed form of the Fibonacci series again.
- We leave out some of the more subtle mathematical aspects, such as the question of convergence of the power series used.

2 / 30

Fibonacci Series – Generating Functions

D2. Advanced Methods for Recurrences

Fibonacci Series – Generating Functions

Power Series

Definition (power series) Let $(a_n)_{n \in \mathbb{N}_0}$ be a sequence of real numbers. The power series with coefficients (a_n) is the (possibly partial) function $g : \mathbb{R} \to \mathbb{R}$ defined by

 $g(x) = \sum_{n=0}^{\infty} a_n x^n$ for all $x \in \mathbb{R}$.

German: Potenzreihe

Notes: more general definitions exist, for example

- using $(x c)^n$ instead of x^n for some $c \in \mathbb{R}$
- using complex instead of real numbers
- using multiple variables

Malte Helmert, Gabriele Röger (University of Discrete Mathematics in Computer Science

D2. Advanced Methods for Recurrences **Uniqueness of Power Series Representation** Theorem Let g and h be power series with coefficients (a_n) and (b_n) . Let $\varepsilon > 0$ such that for all $|x| < \varepsilon$: \bullet g and h converge, and \bullet g(x) = h(x). Then $a_n = b_n$ for all $n \in \mathbb{N}_0$.

```
D2. Advanced Methods for Recurrences

Power Series – Examples

Reminder: g(x) = \sum_{n=0}^{\infty} a_n x^n

Examples:

• a_n = 1

\Rightarrow g(x) = \frac{1}{1-x} (only defined for |x| < 1)

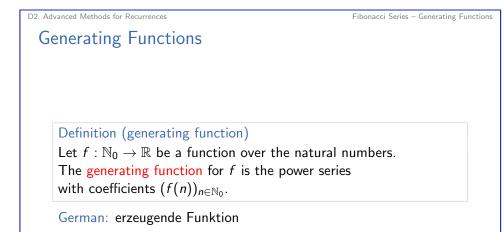
• a_n = z^n for some z \in \mathbb{R}

\Rightarrow g(x) = \frac{1}{1-zx} (only defined for |x| < 1/|z|)

• a_n = \frac{1}{1-zx} (only defined for all x)

• a_n = \begin{cases} 0 & x \text{ is even} \\ \frac{(-1)^{(n-1)/2}}{n!} & x \text{ is odd} \\ \Rightarrow g(x) = \sin x \text{ (defined for all } x) \end{cases}
```

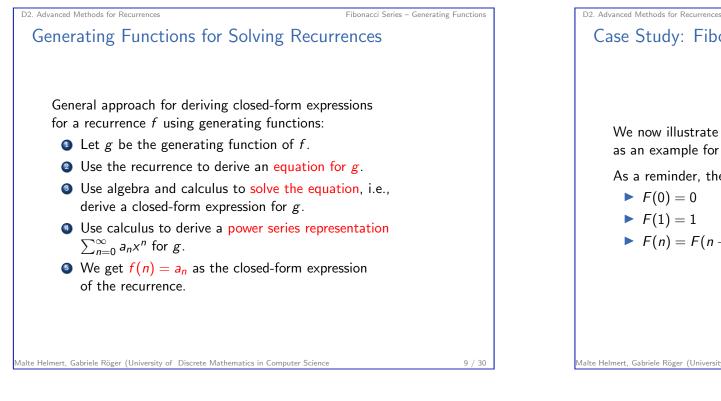
Alte Helmert, Gabriele Röger (University of Discrete Mathematics in Computer Science



We are particularly interested in the case where f is defined by a recurrence.

Malte Helmert, Gabriele Röger (University of Discrete Mathematics in Computer Science

5 / 30



D2. Advanced Methods for Recurrences

Fibonacci Series – Generating Functions

Case Study: 1. Generating Function

1. Let g be the generating function of f.

$$g(x)=\sum_{n=0}^{\infty}F(n)x^n$$
 for $x\in\mathbb{R}$

Note: The series does not converge for all x, but it converges for $|x| < \varepsilon$ for sufficiently small $\varepsilon > 0$. We omit details.

Case Study: Fibonacci Numbers

We now illustrate the approach using the Fibonacci numbers Fas an example for the recurrence f.

As a reminder, the Fibonacci numbers are defined as follows:

► F(n) = F(n-1) + F(n-2) for all n > 2

Ialte Helmert, Gabriele Röger (University of Discrete Mathematics in Computer Science

D2. Advanced Methods for Recurrences Fibonacci Series – Generating Functions Case Study: 2. Equation for g from Recurrence F(0) = 0F(1) = 1F(n) = F(n-1) + F(n-2) for all $n \ge 2$ 2. Use the recurrence to derive an equation for g $g(x) = \sum_{n=0}^{\infty} F(n)x^n = 0 \cdot x^0 + 1 \cdot x^1 + \sum_{n=0}^{\infty} (F(n-1) + F(n-2))x^n$ $= x + \sum_{n=2}^{\infty} F(n-1)x^n + \sum_{n=2}^{\infty} F(n-2)x^n$ $= x + \sum_{n=1}^{\infty} F(n)x^{n+1} + \sum_{n=1}^{\infty} F(n)x^{n+2}$ $= x + x \sum_{n=1}^{\infty} F(n)x^n + x^2 \sum_{n=0}^{\infty} F(n)x^n$ $= x + x \sum_{n=0}^{\infty} F(n)x^n + x^2 \sum_{n=0}^{\infty} F(n)x^n$ $= x + x g(x) + x^2 g(x)$

Fibonacci Series – Generating Functions

Case Study: 3. Solve Equation for g

3. Use algebra and calculus to solve the equation, i.e., derive a closed-form expression for g.

$$g(x) = x + x g(x) + x^2 g(x)$$

$$\Rightarrow \quad g(x) - x g(x) - x^2 g(x) = x$$

$$\Rightarrow \quad g(x)(1 - x - x^2) = x$$

$$\Rightarrow \quad g(x) = \frac{x}{1 - x - x^2}$$

Malte Helmert, Gabriele Röger (University of Discrete Mathematics in Computer Science

122. Advanced Methods for Recurrences Case Study: 4. Power Series Representation for g (2) 4. Use calculus to derive a power series representation $\sum_{n=0}^{\infty} a_n x^n$ for g. (1) a + b = 1, (2) $-a\beta - b\alpha = 0$, (3) $-\alpha - \beta = -1$, (4) $\alpha\beta = -1$ From (3): (5) $\beta = 1 - \alpha$ Substituting (5) into (4): $\alpha(1 - \alpha) = -1$ $\Rightarrow \alpha - \alpha^2 = -1$ $\Rightarrow \alpha^2 - \alpha - 1 = 0$ $\Rightarrow \alpha = \frac{1}{2} \pm \sqrt{\frac{1}{4} + 1} = \frac{1}{2} \pm \sqrt{\frac{5}{4}}$ $\Rightarrow \alpha = \frac{1 \pm \sqrt{5}}{2}$ \Rightarrow The solutions are $\alpha = \varphi$ or $\alpha = \psi$ from the previous chapter. Continue with (6) $\alpha = \varphi$.

13 / 30

Case Study: 4. Power Series Representation for g(1)

4. Use calculus to derive a power series representation $\sum_{n=0}^{\infty} a_n x^n$ for g.

 $g(x) = \frac{x}{1-x-x^2} = xh(x)$ with $h(x) = \frac{1}{1-x-x^2}$ Idea: partial fraction decomposition, i.e., find a, b, α, β such that $h(x) = \frac{a}{1-\alpha x} + \frac{b}{1-\beta x}$.

$$\frac{a}{1-\alpha x} + \frac{b}{1-\beta x} = \frac{a(1-\beta x) + b(1-\alpha x)}{(1-\alpha x)(1-\beta x)}$$
$$= \frac{a-a\beta x + b - b\alpha x}{1-\alpha x - \beta x + \alpha \beta x^2}$$
$$= \frac{(a+b) + (-a\beta - b\alpha)x}{1+(-\alpha - \beta)x + \alpha \beta x^2}$$
$$\Rightarrow a+b=1, \quad -a\beta - b\alpha = 0, \quad -\alpha - \beta = -1, \quad \alpha\beta = -1$$

Malte Helmert, Gabriele Röger (University of Discrete Mathematics in Computer Science

D2. Advanced Methods for Recurrences
Fibonacci Series 7 Generating Functions
Case Study: 4. Power Series Representation for g (3)
4. Use calculus to derive a power series representation
$$\sum_{n=0}^{\infty} a_n x^n$$
 for g.
(1) $a + b = 1$, (2) $-a\beta - b\alpha = 0$, (3) $-\alpha - \beta = -1$, (4) $\alpha\beta = -1$,
(5) $\beta = 1 - \alpha$, (6) $\alpha = \varphi$
Substituting (6) into (5): (7) $\beta = 1 - \alpha = 1 - \varphi = \psi$.
From (1): (8) $b = 1 - a$
Substituting (6), (7), (8) into (2):
 $-a(1 - \varphi) - (1 - a)\varphi = 0$
 $\Rightarrow -a + a\varphi - \varphi + a\varphi = 0$
 $\Rightarrow a(2\varphi - 1) = \varphi$
 $\Rightarrow a = \frac{\varphi}{2\varphi - 1} = \frac{\varphi}{2 \cdot \frac{1}{2}(1 + \sqrt{5}) - 1} = \frac{1}{\sqrt{5}}\varphi$

Case Study: 4. Power Series Representation for g(4)

4. Use calculus to derive a power series representation $\sum_{n=0}^{\infty} a_n x^n$ for g. (8) b = 1 - a, (9) $a = \frac{1}{\sqrt{5}}\varphi$

Substituting (9) into (8):

$$b = 1 - a$$

$$= 1 - \frac{1}{\sqrt{5}}\varphi$$

$$= \frac{\sqrt{5}}{\sqrt{5}} - \frac{\frac{1}{2}(1 + \sqrt{5})}{\sqrt{5}}$$

$$= -\frac{1}{\sqrt{5}}(-\sqrt{5} + \frac{1}{2} + \frac{1}{2}\sqrt{5})$$

$$= -\frac{1}{\sqrt{5}}(\frac{1}{2} - \frac{1}{2}\sqrt{5})$$

$$= -\frac{1}{\sqrt{5}}\psi$$
Solviele Biger (University of Discrete Mathematics in Computer Science 17 / 30

D2. Advanced Methods for Recurrences

Aalte Helmert

Fibonacci Series – Generating Functions

Case Study: 5. Extract Closed Form of Recurrence

Use calculus to derive a power series representation ∑_{n=0}[∞] a_nxⁿ for g.
 We get f(n) = a_n as the closed-form expression of the recurrence.

From

$$g(x) = \sum_{n=0}^{\infty} \frac{1}{\sqrt{5}} (\varphi^n - \psi^n) x^n$$

we conclude:

$$F(n)=rac{1}{\sqrt{5}}(arphi^n-\psi^n) ext{ for all } n\in\mathbb{N}_0$$

D2. Advanced Methods for Recurrences

Malte H

Fibonacci Series - Generating Functions

Case Study: 4. Power Series Representation for g (5)

4. Use calculus to derive a power series representation $\sum_{n=0}^{\infty} a_n x^n$ for g. $g(x) = xh(x), \quad h(x) = \frac{a}{1-\alpha x} + \frac{b}{1-\beta x},$ $\alpha = \varphi, \quad \beta = \psi, \quad a = \frac{1}{\sqrt{5}}\varphi, \quad b = -\frac{1}{\sqrt{5}}\psi$

Plugging everything in:

$$g(x) = x \left(\frac{1}{\sqrt{5}}\varphi \frac{1}{1-\varphi x} - \frac{1}{\sqrt{5}}\psi \frac{1}{1-\psi x}\right) = \frac{x}{\sqrt{5}} \left(\varphi \frac{1}{1-\varphi x} - \psi \frac{1}{1-\psi x}\right)$$
$$= \frac{x}{\sqrt{5}} \left(\varphi \sum_{n=0}^{\infty} \varphi^n x^n - \psi \sum_{n=0}^{\infty} \psi^n x^n\right)$$
$$= \frac{1}{\sqrt{5}} \left(\sum_{n=0}^{\infty} \varphi^{n+1} x^{n+1} - \sum_{n=0}^{\infty} \psi^{n+1} x^{n+1}\right)$$
$$= \frac{1}{\sqrt{5}} \left(\sum_{n=1}^{\infty} \varphi^n x^n - \sum_{n=1}^{\infty} \psi^n x^n\right) = \sum_{n=1}^{\infty} \frac{1}{\sqrt{5}} (\varphi^n - \psi^n) x^n$$
$$= \sum_{n=0}^{\infty} \frac{1}{\sqrt{5}} (\varphi^n - \psi^n) x^n$$

D2. Advanced Methods for Recurrences Fibonacci Series - Generating Functions Concluding Remarks The approach requires analytical skill, but once understood, it can be applied to many similar recurrences. The same basic idea can be used to solve all recurrences of the form f(0) = a₀ ... f(k-1) = a_{k-1} f(n) = c₁f(n-1) + ... + c_kf(n-k) for all n ≥ k The Fibonacci numbers are the special case where k = 2, a₀ = 0, a₁ = 1, c₁ = 1, c₂ = 1.

D2.2 Master Theorem for Divide-and-Conquer Recurrences

Malte Helmert, Gabriele Röger (University of Discrete Mathematics in Computer Science

D2. Advanced Methods for Recurrences

Master Theorem for Divide-and-Conquer Recurrences

Asymptotic Growth

- Run-time analysis usually focuses on the asymptotic growth rate of run-time.
- For example, we say "run-time grows at most quadratically" rather than saying that run-time for inputs of size n is 3n² + 17n + 8.

advantages:

- much simpler to study
- can abstract from minor implementation details

Master Theorem for Divide-and-Conquer Recurrences

Divide-and-Conquer Algorithms

Recurrences frequently arise in the run-time analysis of divide-and-conquer algorithms.

► Examples:

- Mergesort: sort a sequence by recursively sorting two smaller sequences, then merging them
- Binary search: find an element in a sorted sequence by identifying which half of the sequence must contain the element, then recursively searching it
- Quickselect: find the k-th smallest element in a sequence by recursive partitioning

Alte Helmert, Gabriele Röger (University of Discrete Mathematics in Computer Science

22 / 30

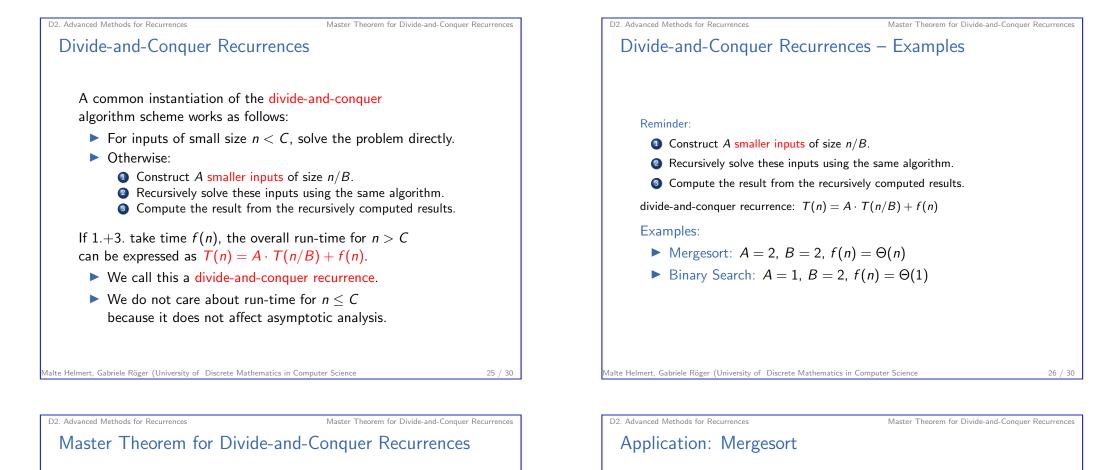
D2. Advanced Methods for Recurrences Big-O, Big- Ω , Big- Θ

Definition (O, Ω, Θ)

Let $g : \mathbb{R}_0^+ \to \mathbb{R}$ be a function. The sets of functions $O(g), \Omega(g), \Theta(g)$ are defined as follows: $\bullet \quad O(g) = \{f : \mathbb{R}_0^+ \to \mathbb{R} \mid \text{there exist } C, n_0 \in \mathbb{R}$ $\text{ s.t. } |f(n)| \le C \cdot g(n) \text{ for all } n \ge n_0\}$ $\bullet \quad \Omega(g) = \{f : \mathbb{R}_0^+ \to \mathbb{R} \mid \text{there exist } C, n_0 \in \mathbb{R}$ $\text{ s.t. } |f(n)| \ge C \cdot g(n) \text{ for all } n \ge n_0\}$ $\bullet \quad \Theta(g) = O(g) \cap \Omega(g)$

Notation:

- ► It is convention to say $"5n^2 + 7n \log_2 n = \Theta(n^2)"$ instead of " $f \in \Theta(g)$ for the functions f, gwith $f(n) = 5n^2 + 7n \log_2 n$ and $g(n) = n^{2"}$.
- ditto for O, Ω



Let $A \ge 1$, $B \ge 1$, and let T satisfy the divide-and-conquer recurrence $T(n) = A \cdot T(n/B) + f(n)$. Then:

- ► If $f(n) = O(n^{\log_B A \varepsilon})$ for some $\varepsilon > 0$, then $T(n) = \Theta(n^{\log_B A})$.
- ► If $f(n) = \Theta(n^{\log_B A})$, then $T(n) = \Theta(n^{\log_B A} \log_2 n)$.
- If $f(n) = \Omega(n^{\log_B A + \varepsilon})$ for some $\varepsilon > 0$, then $T(n) = \Theta(f(n))$.

We do not prove the theorem.

Reminder: $T(n) = A \cdot T(n/B) + f(n)$

 $\blacktriangleright f(n) = O(n^{\log_B A - \varepsilon}) \rightsquigarrow T(n) = \Theta(n^{\log_B A})$

 $\blacktriangleright f(n) = \Omega(n^{\log_B A + \varepsilon}) \rightsquigarrow T(n) = \Theta(f(n))$

Mergesort: A = 2, B = 2, $f(n) = \Theta(n)$

 $\blacktriangleright f(n) = O(n^{1-\varepsilon}) \rightsquigarrow T(n) = \Theta(n^1)$

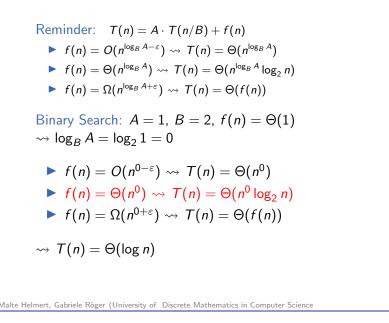
 $\blacktriangleright f(n) = \Theta(n^1) \rightsquigarrow T(n) = \Theta(n^1 \log_2 n)$

 $\blacktriangleright f(n) = \Omega(n^{1+\varepsilon}) \rightsquigarrow T(n) = \Theta(f(n))$

 $\rightsquigarrow \log_B A = \log_2 2 = 1$

 $\rightsquigarrow T(n) = \Theta(n \log n)$

 $f(n) = \Theta(n^{\log_B A}) \rightsquigarrow T(n) = \Theta(n^{\log_B A} \log_2 n)$



29 / 30

D2. Advanced Methods for Recurrences

More Complex Cases

Some divide-and-conquer algorithms have more complicated recurrences because they do not split into even-sized pieces of predictable size.

Example:

- Quicksort with random pivotization: f(n) = Θ(n); split n uniformly randomly into 1 ≤ k ≤ n and n − 1 − k ~ expected runtime Θ(n log n)
- Quickselect with median-of-median pivotization: f(n) = Θ(n); one recursion on input size n/5, one recursion on input size at most n ⋅ ⁷/₁₀ → runtime Θ(n)

Here, we can try to use the Master theorem to derive hypotheses and then prove them by mathematical induction.

Alte Helmert, Gabriele Röger (University of Discrete Mathematics in Computer Science