

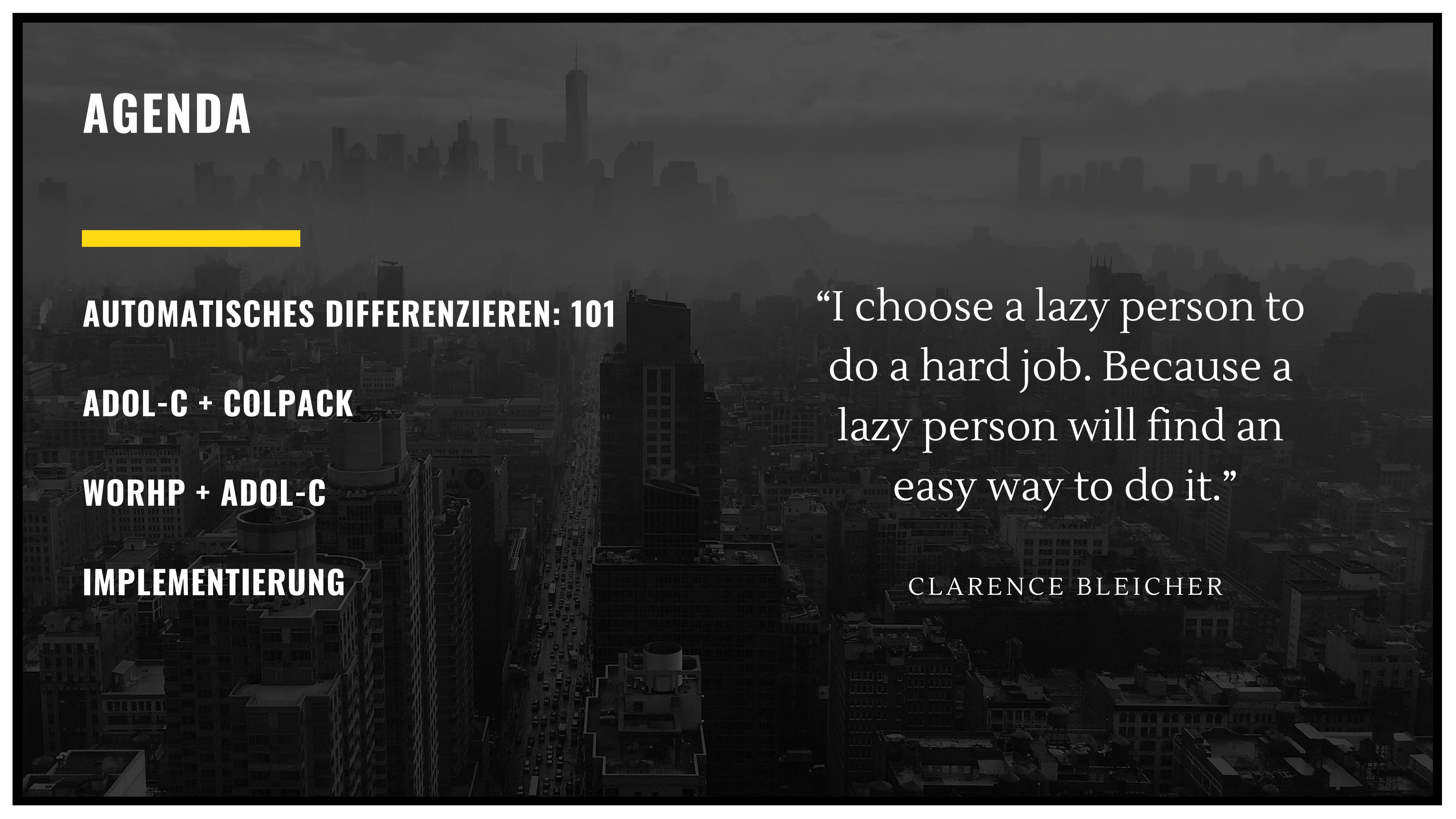


AUTOMATISCHES DIFFERENZIEREN IN WORHP

WORHP =>> ADOL-C

LUIS LÜTTGENS

AGENDA



AUTOMATISCHES DIFFERENZIEREN: 101

ADOL-C + COLPACK

WORHP + ADOL-C

IMPLEMENTIERUNG

“I choose a lazy person to
do a hard job. Because a
lazy person will find an
easy way to do it.”

CLARENCE BLEICHER

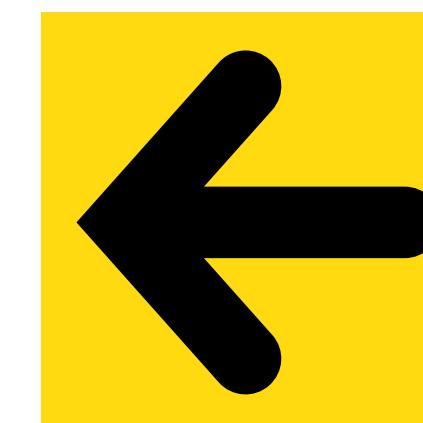
Automatisches Differenzieren: 101



AUTOMATISCHES ABLEITEN VON ALGORITHMEN



ALGORITHMUS ALS BINÄRBAUM DARSTELLEN

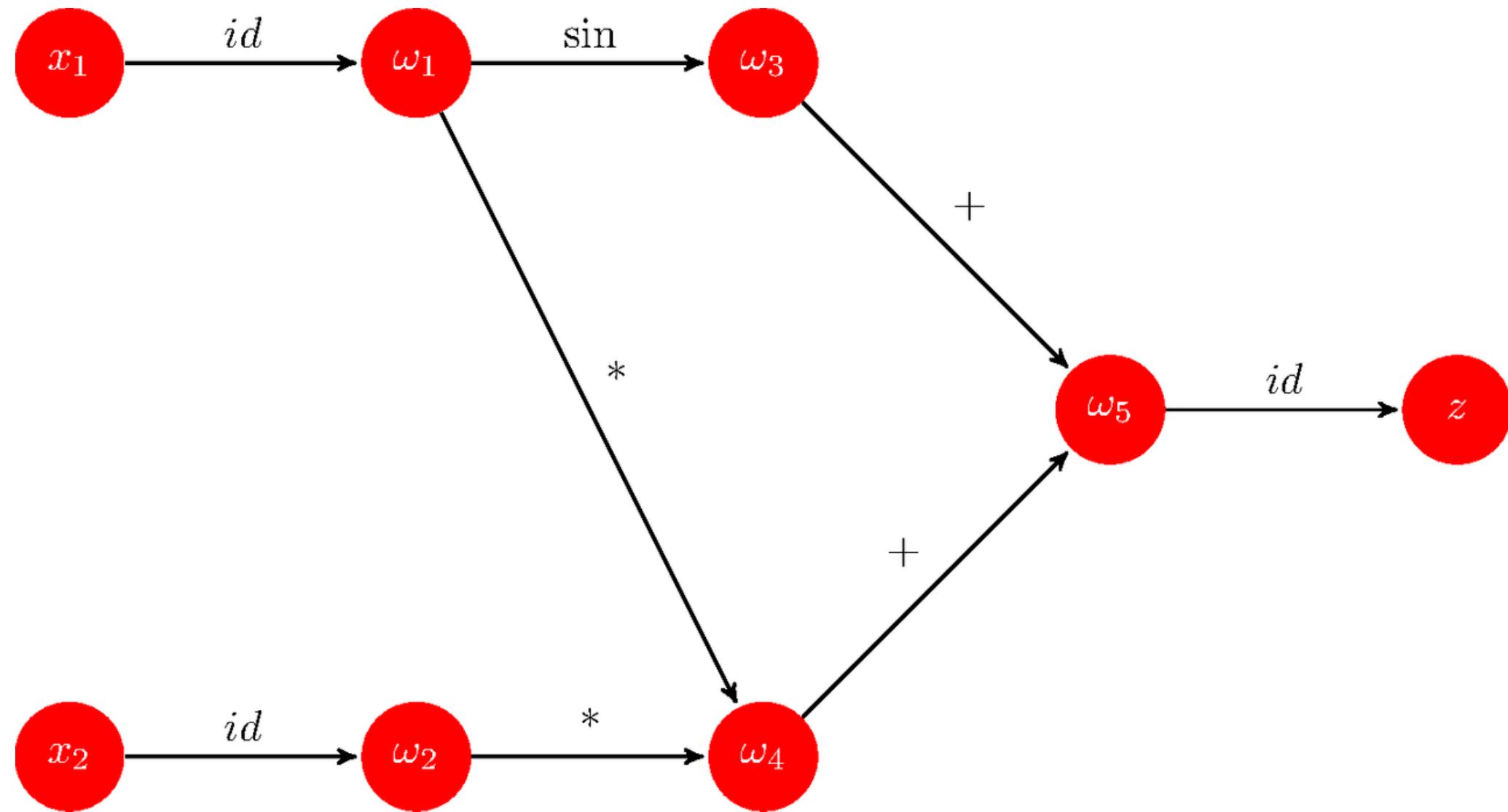


RÜCKWÄRTS DURCH DEN BAUM WANDERN MIT
KETTENREGEL

Beispiel

$$\underline{z(x_1, x_2) = x_1 * x_2 + \sin(x_1)}$$

Forward pass:



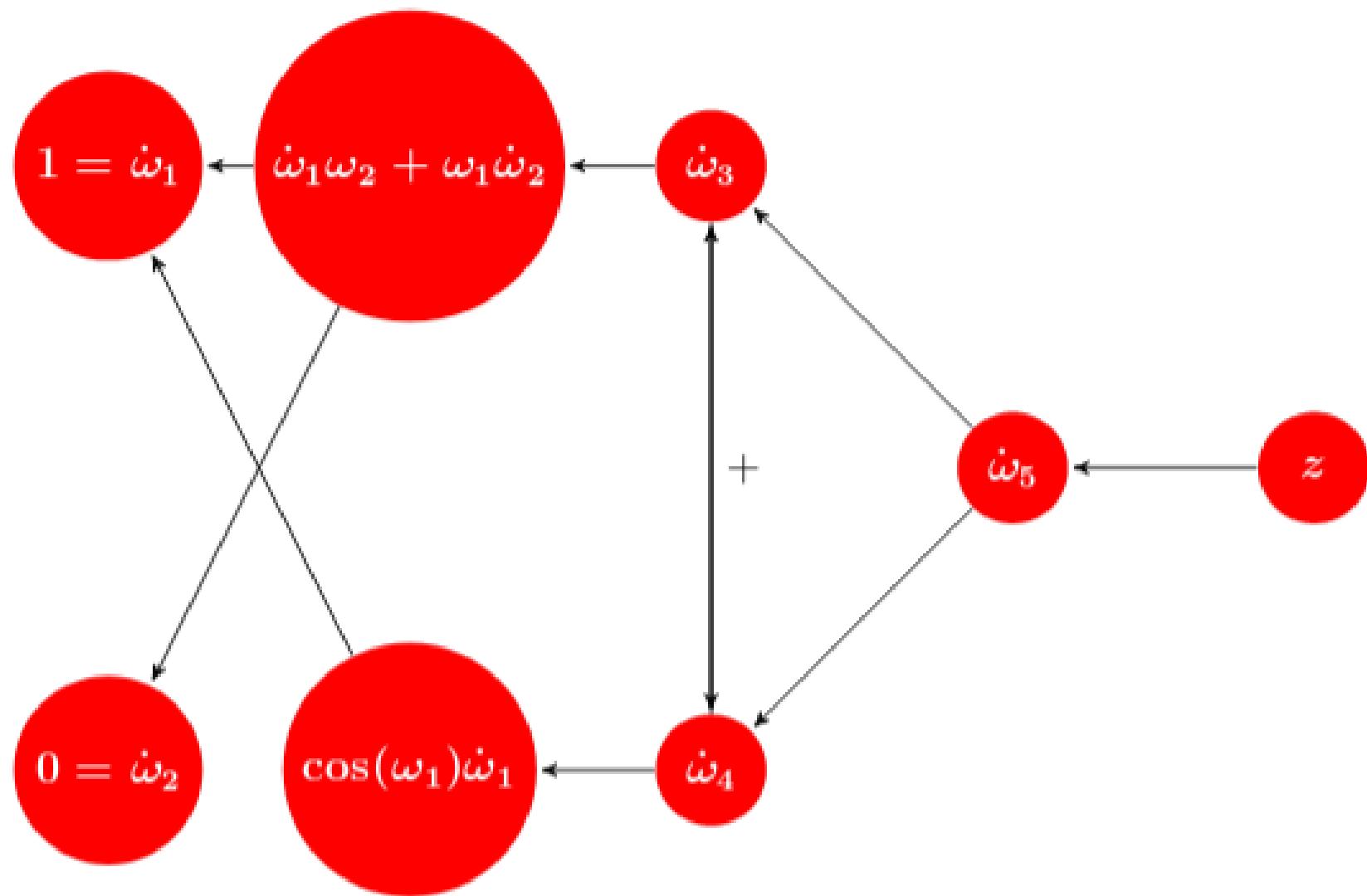
Auswertung der Symbole:

ω_1	$=$	x_1	$=$	2
ω_2	$=$	x_2	$=$	3
ω_3	$=$	$\omega_1 \omega_2$	$=$	6
ω_4	$=$	$\sin(\omega_1)$	$=$	0.9
ω_5	$=$	$\omega_3 + \omega_4$	$=$	6.9
z	$=$	ω_5	$=$	6.9

Beispiel

$$\underline{z(x_1, x_2) = x_1 * x_2 + \sin(x_1)}$$

Backward pass:



Auswertung der Ableitung:

$$\dot{\omega}_1 = \frac{dx_1}{dx_1} = 1$$

$$\dot{\omega}_2 = \frac{dx_2}{dx_1} = 0$$

$$\dot{\omega}_3 = \dot{\omega}_1\dot{\omega}_2 + \omega_1\dot{\omega}_2$$

$$\dot{\omega}_4 = \cos(\omega_1)\dot{\omega}_1$$

$$\dot{\omega}_5 = \dot{\omega}_3 + \dot{\omega}_4$$

$$\frac{dz}{dx_1} = \dot{\omega}_5$$

ADOL-C + COLPACK

ADOL-C UNTERSTÜTZT EXPLIZIT
OPTIMIERUNGSPROBLEME

EASY-TO-USE DRIVERS
LIEFERN JACOBI- UND HESSEMATRIX

COLPACK BERECHNET ZUSÄTZLICH DIE
DÜNNBESETZTHEIT

WORHP + ADOL-C

Was muss getan werden?

I.

II.

III.

Templates

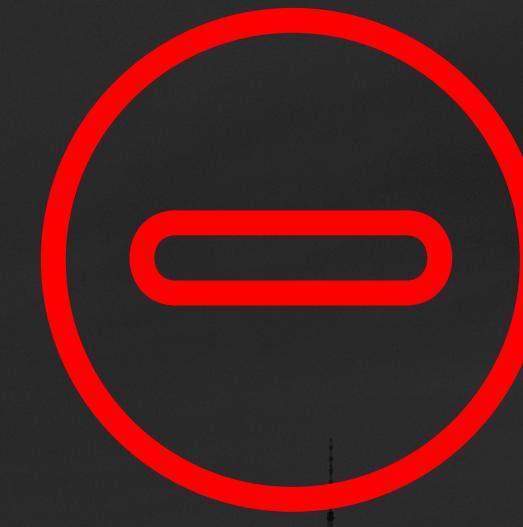
UserF und UserG werden aktiv

Init ADOL-C

Generation der tapes

Interface

Zusammenspiel von WORHP und
ADOL-C



RAPID PROTOTYPING

KEINE
FLÜCHTIGKEITSFEHLER

EVTL. PERFORMANCE
EINBUSSEN