Regular Linear Temporal Logic

Malte Schmitz

Universität zu Lübeck

20. Januar 2011

Ziele

1. Schwäche von LTL gegenüber reg. Ausdrücken erkennen
2. RLTL kennen lernen
3. die Power-Operatoren verstehen
4. Umwandlung von ω-reg. Ausdrücken und LTL zu RLTL sehen

Dieser Vortrag basiert auf folgender Arbeit:

Martin Leucker and César Sánchez
»Regular Linear Temporal Logic«
In: Jones, C. B., Liu, Z., Woodcock, J. (Hg.) ICTAC 2007.
LNCS, Ausg. 4711, SS. 291–305. Springer, Heidelberg (2007)

- Temporallogik = intuitive Spezifikation zeitlicher Zusammenhänge
- Aussagen werden über unendlichen Bäumen ausgewertet

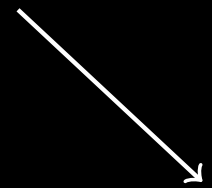
- Linear Temporal Logic (LTL) = Lineare Temporallogik
- Aussagen werden über linearen Pfaden ausgewertet

Linear Temporal Logic (LTL)

$$(\mathsf{N}\,\mathsf{N}\,\mathsf{N}\,x)\,\mathsf{U}\,z$$

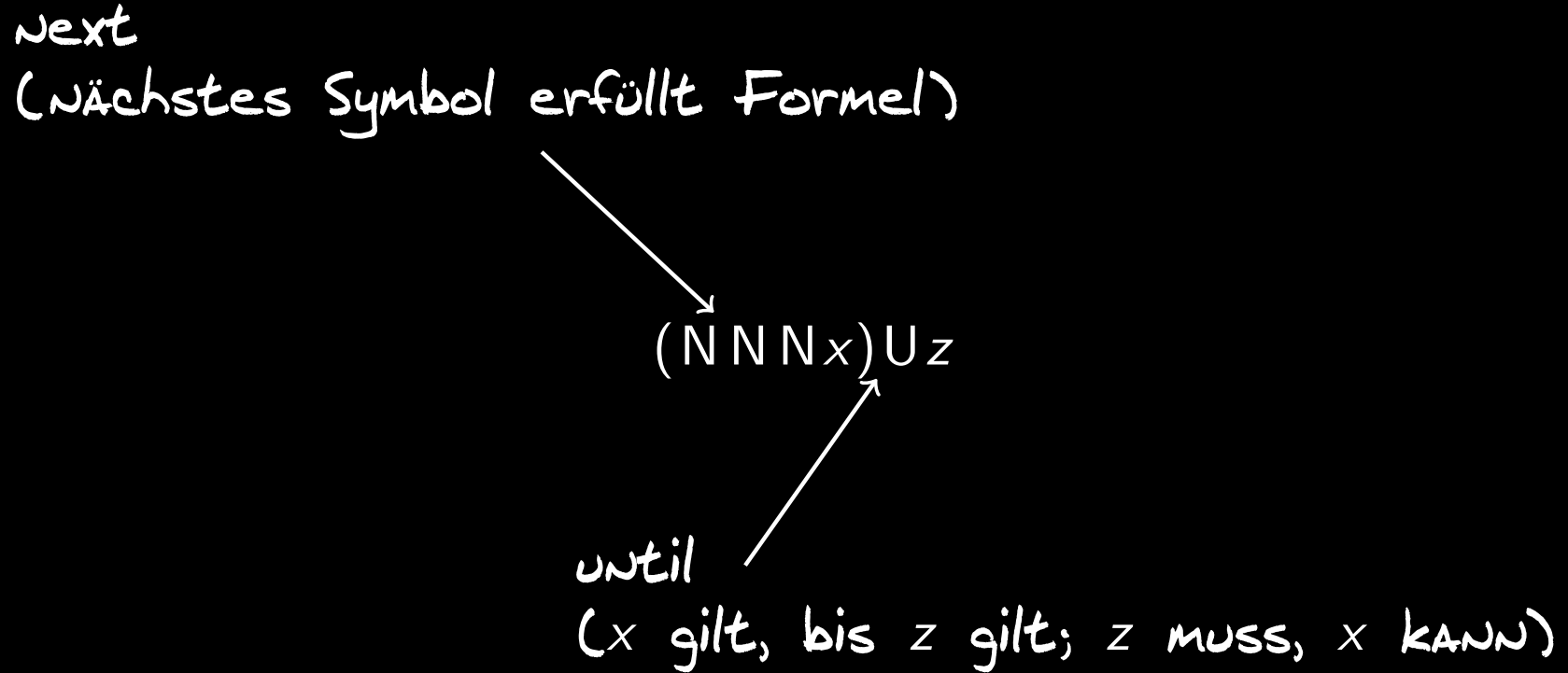
Linear Temporal Logic (LTL)

next
(Nächstes Symbol erfüllt Formel)



$$(\mathsf{N}\,\mathsf{N}\,\mathsf{N}\,x)\,\mathsf{U}\,z$$

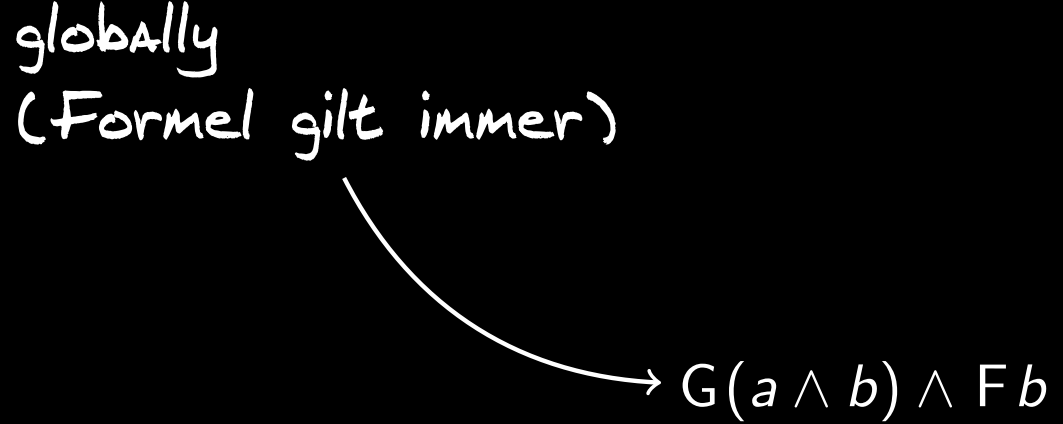
Linear Temporal Logic (LTL)



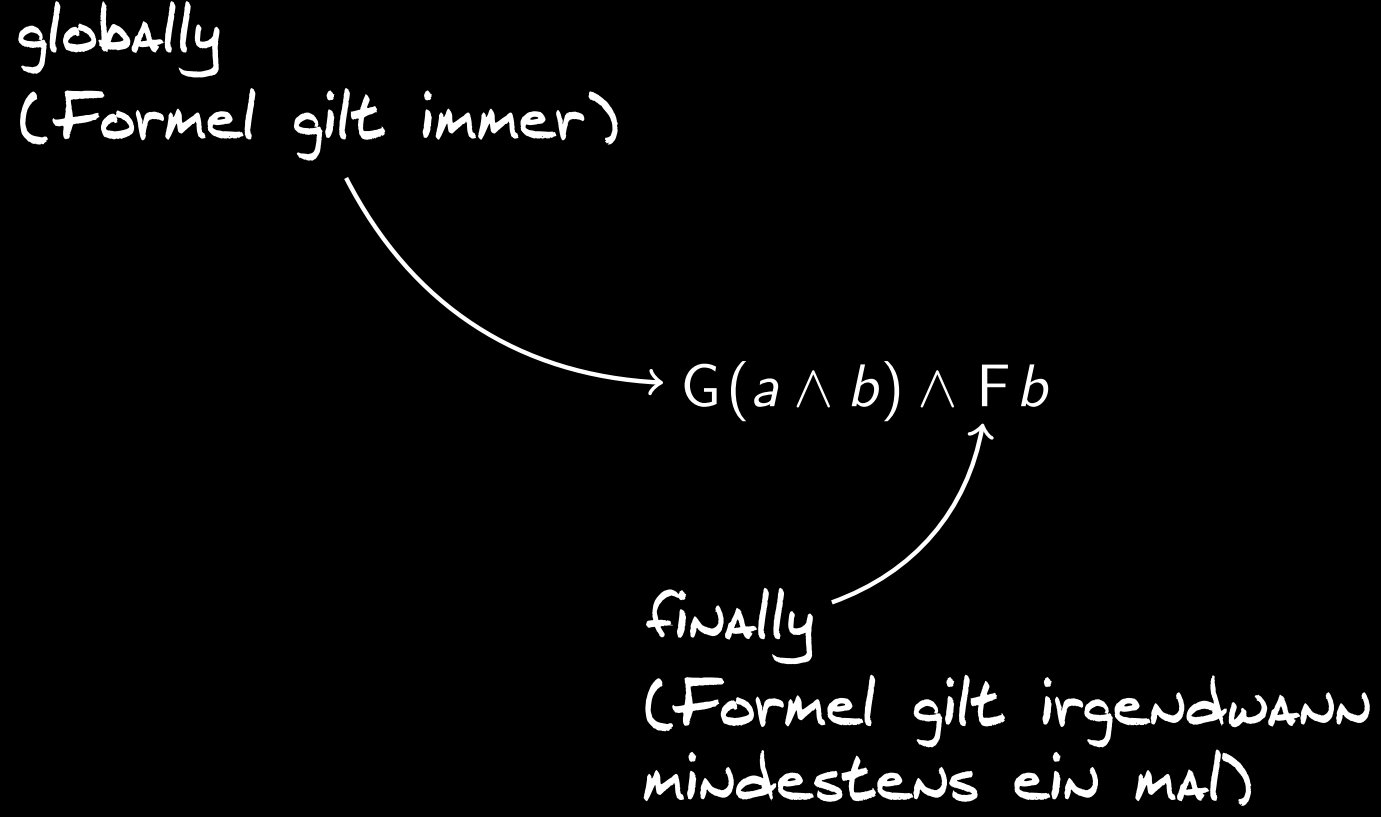
Linear Temporal Logic (LTL)

$$\mathsf{G}(a \wedge b) \wedge \mathsf{F}\, b$$

Linear Temporal Logic (LTL)



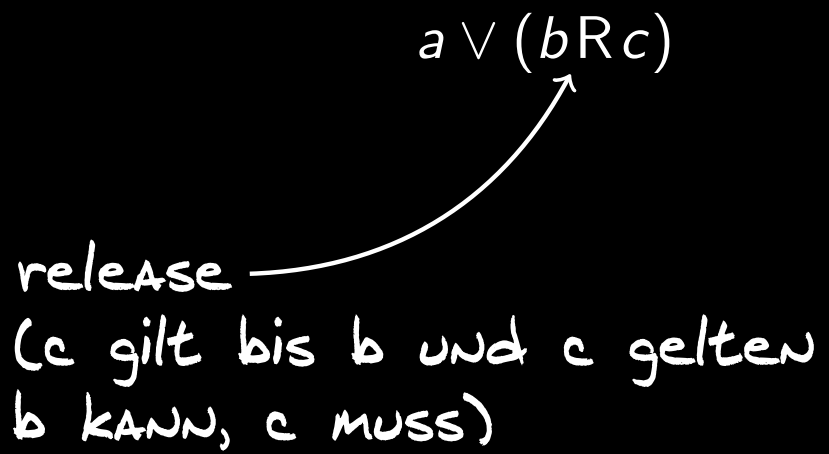
Linear Temporal Logic (LTL)



Linear Temporal Logic (LTL)

$$a \vee (b \mathrm{R} c)$$

Linear Temporal Logic (LTL)



regulärer Ausdruck

$$(x^*x)+(w\,;z)$$

regulärer Ausdruck

binärer Kleene-Stern
endliche Wiederholung!

$$(x^*x)+(w\,;z)$$

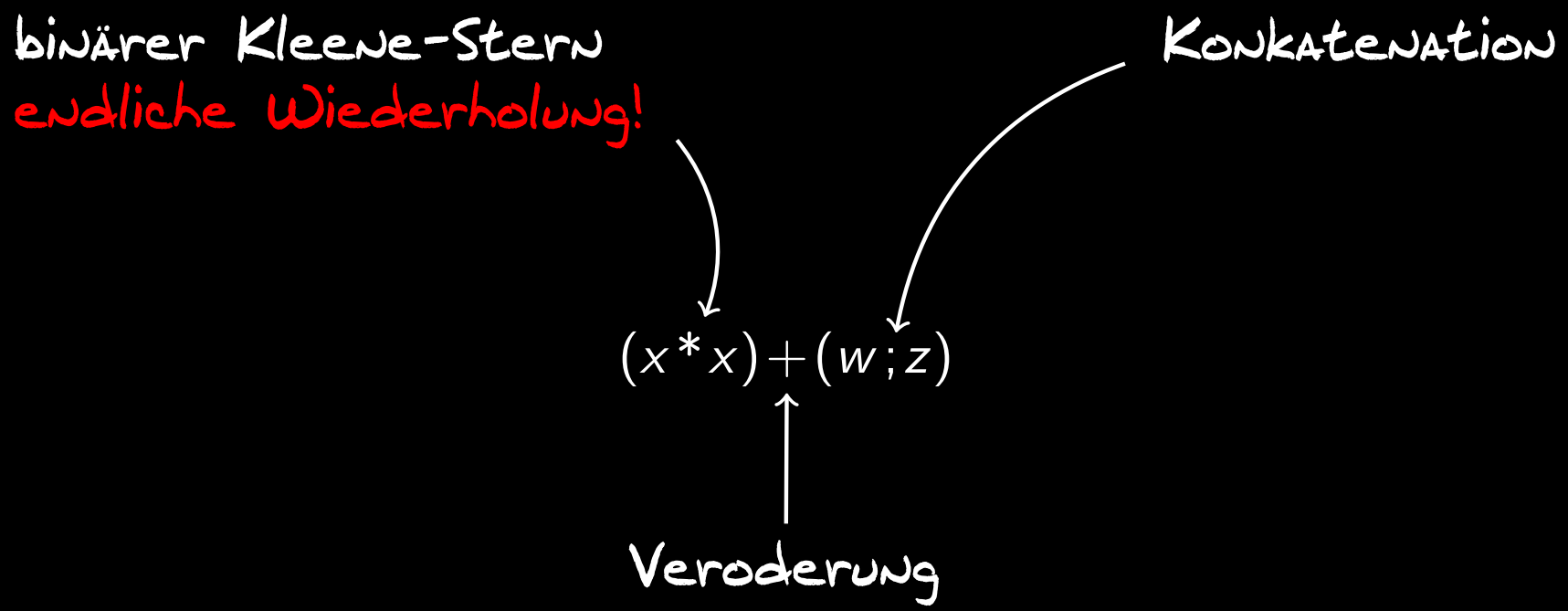
regulärer Ausdruck

binärer Kleene-Stern
endliche Wiederholung!

Konkatenation

$$(x^*x)+(w;z)$$

regulärer Ausdruck

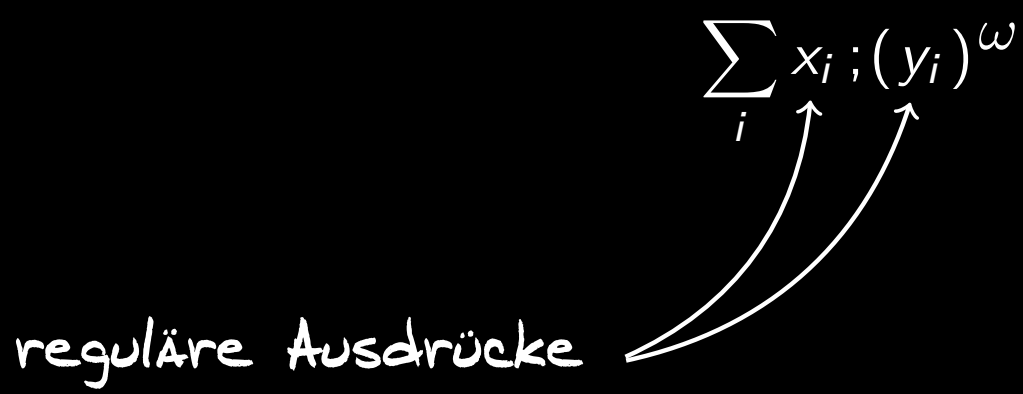


- ω-reguläre Sprache = Menge unendlicher Wörter über gemeinsamen Alphabet
- Jeder ω-reguläre Ausdruck ist von der Form

$$\sum_i x_i \,;(y_i)^\omega$$

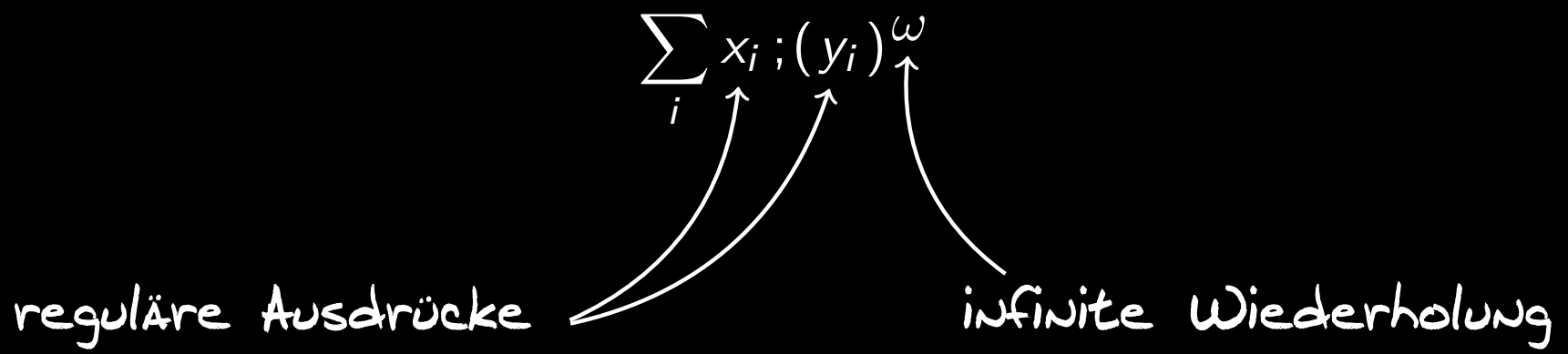
- ω-reguläre Ausdrücke sind wenig intuitiv

- ω-reguläre Sprache = Menge unendlicher Wörter über gemeinsamen Alphabet
- Jeder ω-reguläre Ausdruck ist von der Form



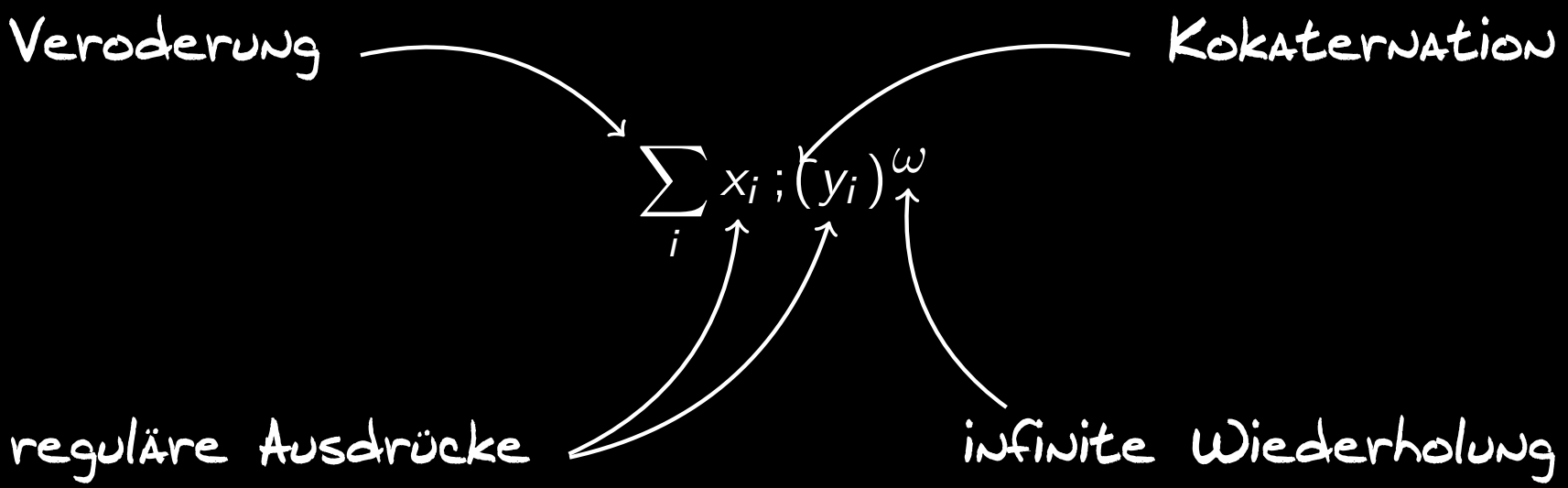
- ω-reguläre Ausdrücke sind wenig intuitiv

- ω-reguläre Sprache = Menge unendlicher Wörter über gemeinsamen Alphabet
- Jeder ω-reguläre Ausdruck ist von der Form



- ω-reguläre Ausdrücke sind wenig intuitiv

- ω-reguläre Sprache = Menge unendlicher Wörter über gemeinsamen Alphabet
- Jeder ω-reguläre Ausdruck ist von der Form



- ω-reguläre Ausdrücke sind wenig intuitiv

LTL

- ist eine sprechende Temporallogik
- beherrscht keine Wiederholungen

RLTL

- kann alle ω-regulären Sprachen definieren
- ist so »schön« wie LTL

Regular Linear Temporal Logic (RLTL)

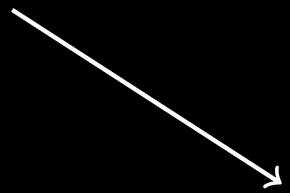
reguläre Ausdrücke
RLTL-Ausdrücke

$$q \vee (\widehat{a * a} \wedge ((a + b); r))$$

Regular Linear Temporal Logic (RLTL)

reguläre Ausdrücke
RLTL-Ausdrücke

Verunendlichung
(endliches Teilwort beliebiger Länge
erfüllt reg. Ausdruck)



$$q \vee (\widehat{a * a} \wedge ((a + b); r))$$

Regular Linear Temporal Logic (RLTL)

reguläre Ausdrücke
RLTL-Ausdrücke

Verunendlichung
(endliches Teilwort beliebiger Länge
erfüllt reg. Ausdruck)

$$q \vee (\widehat{a * a} \wedge ((a + b); r))$$

Konkatenation
(endliches Teilwort beliebiger Länge
erfüllt reg. Ausdruck
und Rest erfüllt RLTL-Ausdruck)

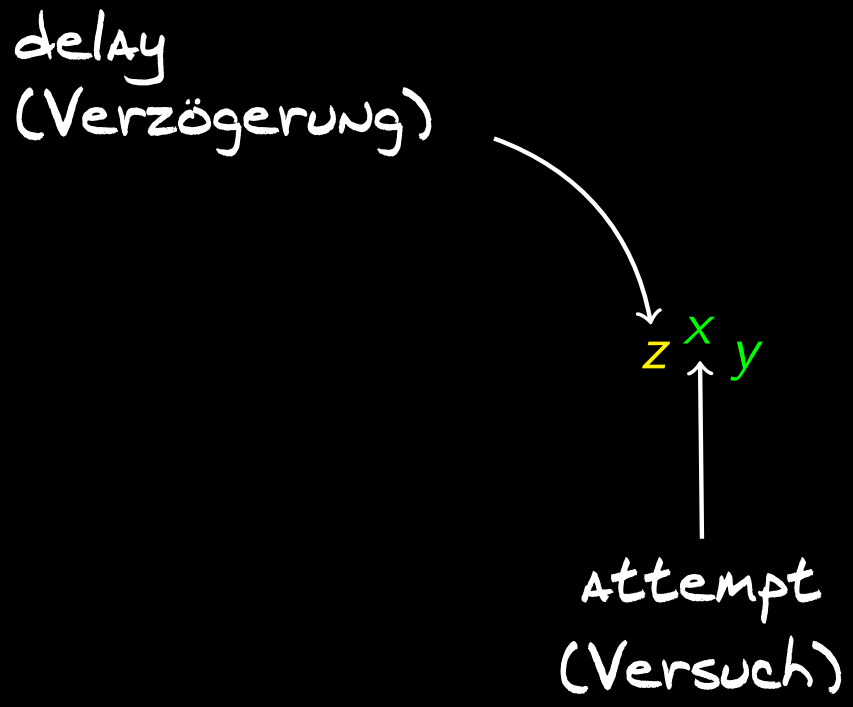
Power-Operator

$$z^{x\,y}$$

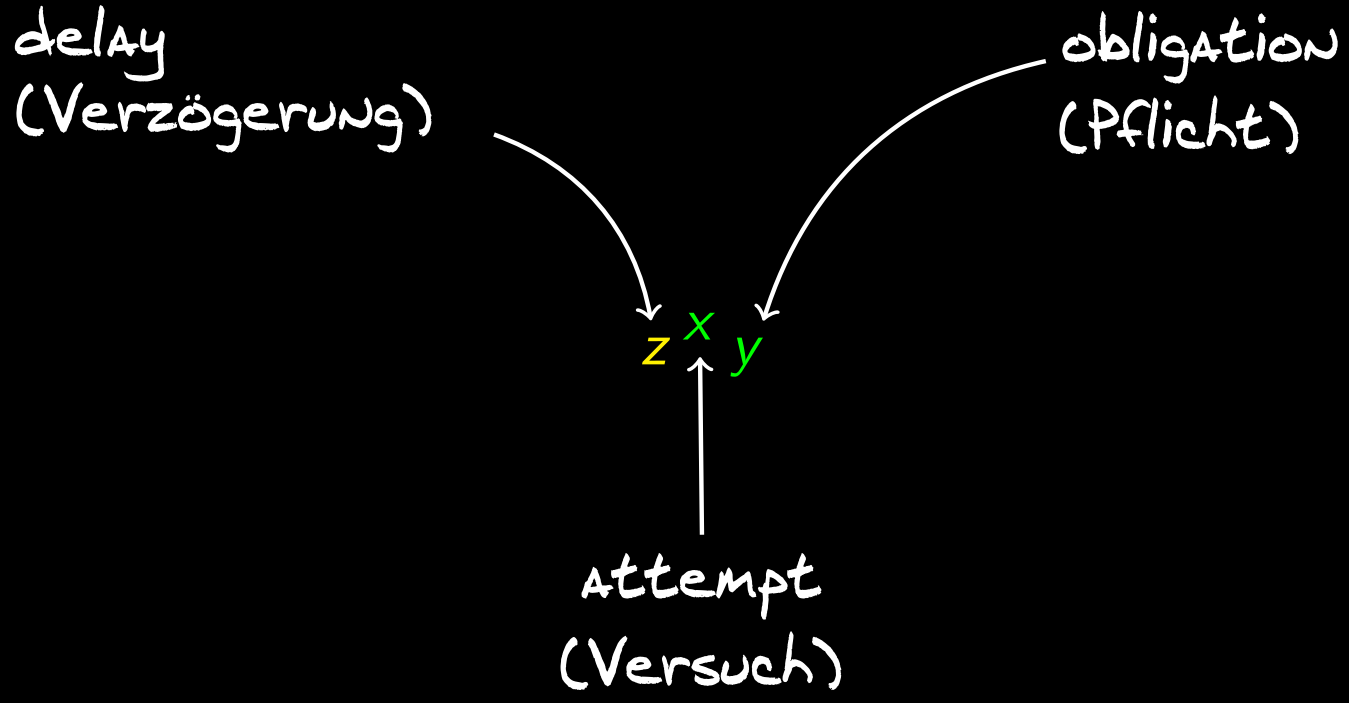
Power-Operator

delay
(Verzögerung)

Power-Operator



Power-Operator



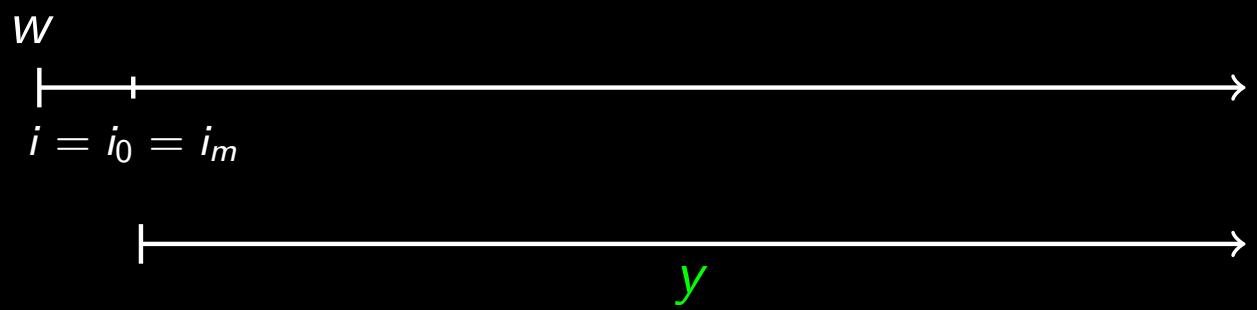
regulärer Ausdruck

RLTL-Ausdruck

$z^{x}\, y$ gilt für w, wenn

– entweder direkt y gilt

– oder immer wieder z und x, bis y gilt



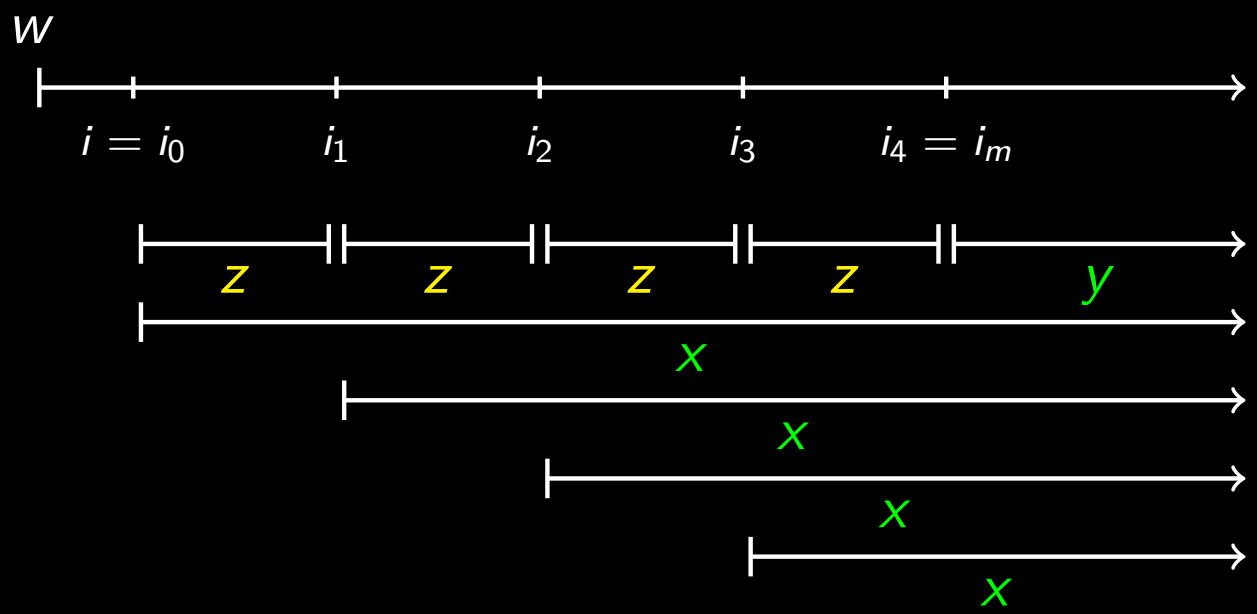
regulärer Ausdruck

RLTL-Ausdruck

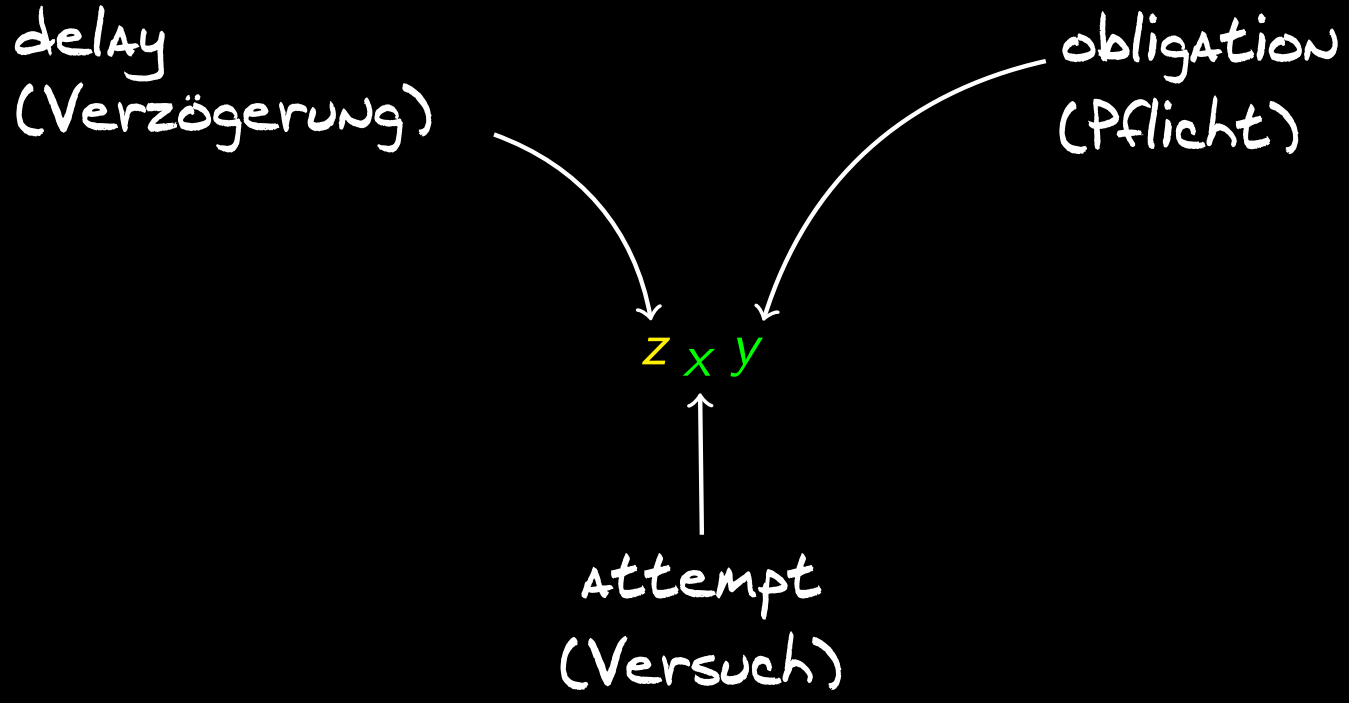
$z^x y$ gilt für w, wenn

– entweder direkt y gilt

– oder mit jedem z immer x, bis y gilt



Power-Operator



regulärer Ausdruck

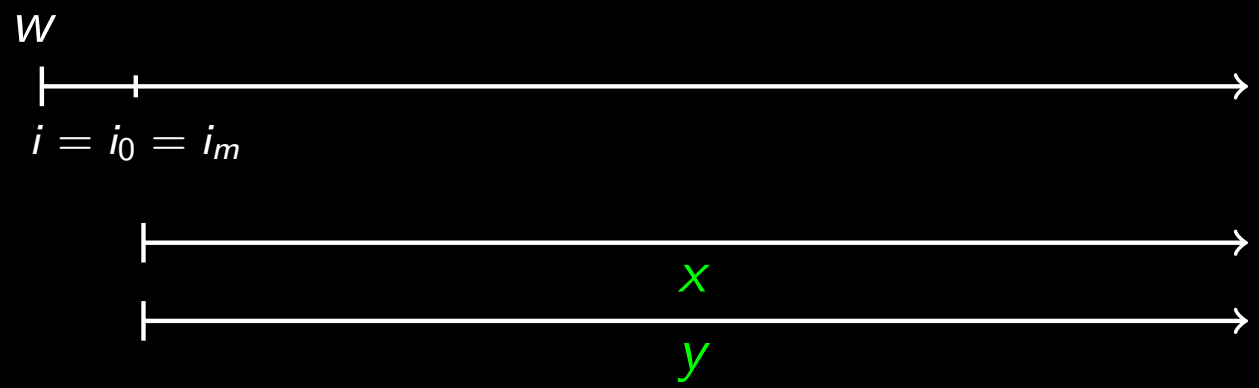
RLTL-Ausdruck

$z\ x\ y$ gilt für w, wenn

– entweder direkt x und y gelten

– oder mit jedem z immer y, bis x und y gelten

– oder unendlich mit jedem z immer y



regulärer Ausdruck

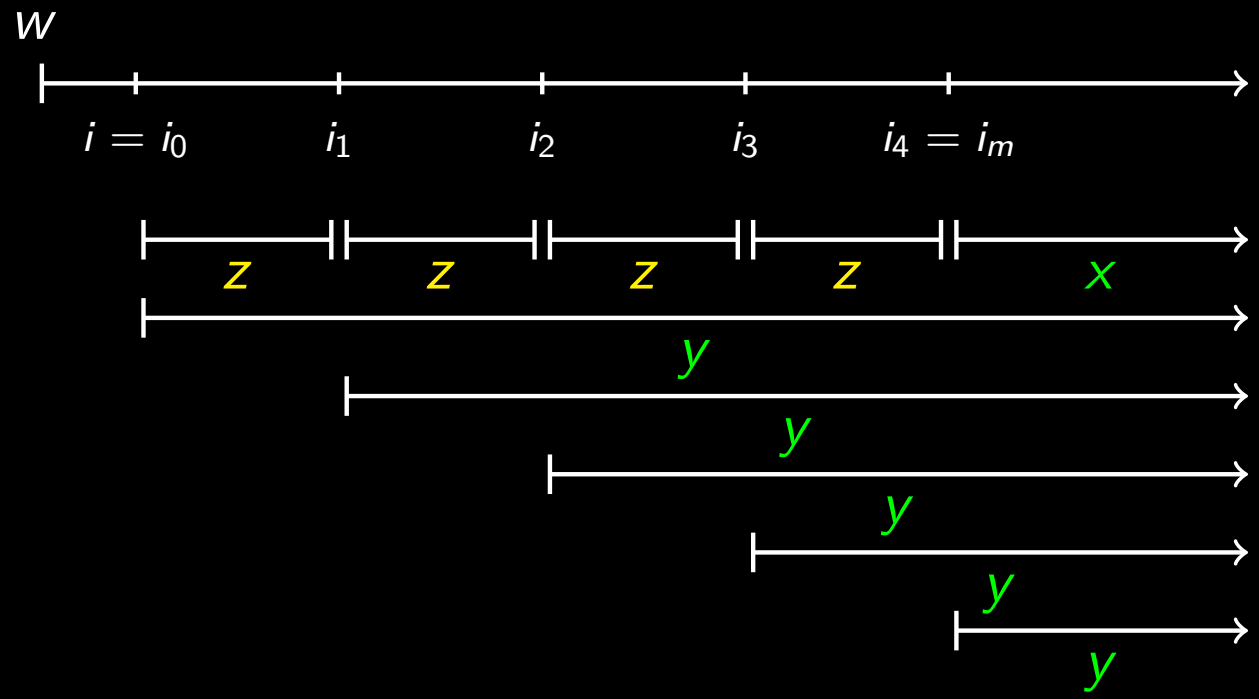
RLTL-Ausdruck

$z\ x\ y$ gilt für w, wenn

– entweder direkt x und y gelten

– oder mit jedem z immer y, bis x und y gelten

– oder unendlich mit jedem z immer y



regulärer Ausdruck

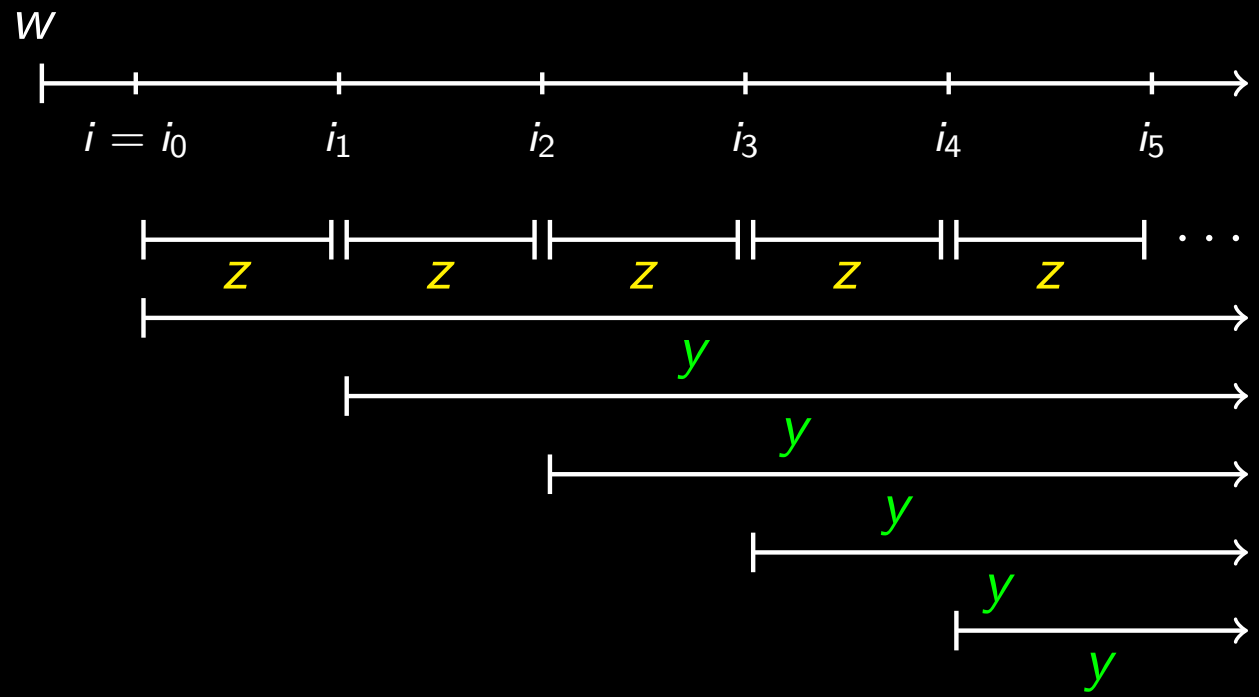
RLTL-Ausdruck

$z\ x\ y$ gilt für w, wenn

– entweder direkt x und y gelten

– oder mit jedem z immer y, bis x und y gelten

– oder unendlich mit jedem z immer y

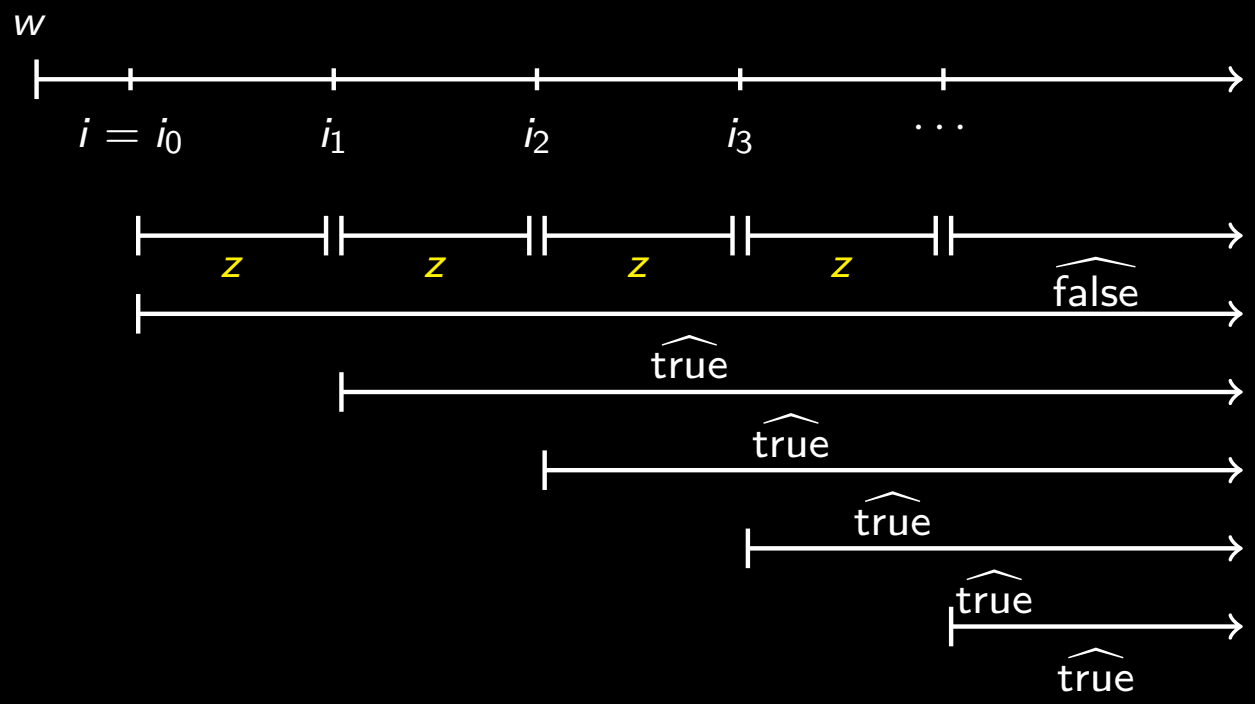


Umwandlung ω-reg. Ausdruck in RLTL

$$z^\omega \quad \equiv \quad z\widehat{\ }\widehat{\mathsf{false}}\widehat{\mathsf{true}}$$

Umwandlung ω-reg. Ausdruck in RLTL

$$z^\omega \quad \equiv \quad z \widehat{\ } \widehat{\text{false}} \widehat{\text{true}}$$



Umwandlung ω-reg. Ausdruck in RLTL

$$z^\omega \quad \equiv \quad z \widehat{\ }_{\mathsf{false}} \widehat{\mathsf{true}}$$

$$\sum_i x_i \,;(y_i)^\omega \quad \equiv \quad \bigvee_i x_i \,;(y_i \widehat{\ }_{\mathsf{false}} \widehat{\mathsf{true}})$$

Umwandlung LTL in RLTL

$$\mathsf{N}\,x \quad \equiv \quad \mathsf{true}\,;x$$

Umwandlung LTL in RLTL

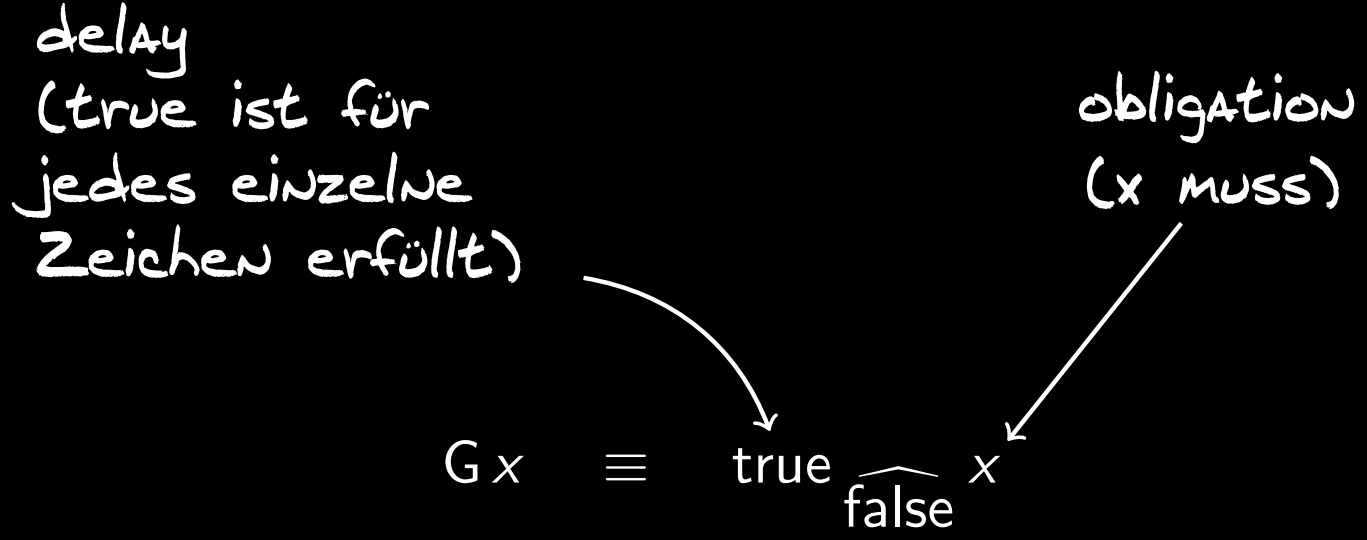
$$\mathsf{G}\,x \quad \equiv \quad \mathsf{true} \overset{\frown}{\underset{\mathsf{false}}{\wedge}} x$$

Umwandlung LTL in RLTL

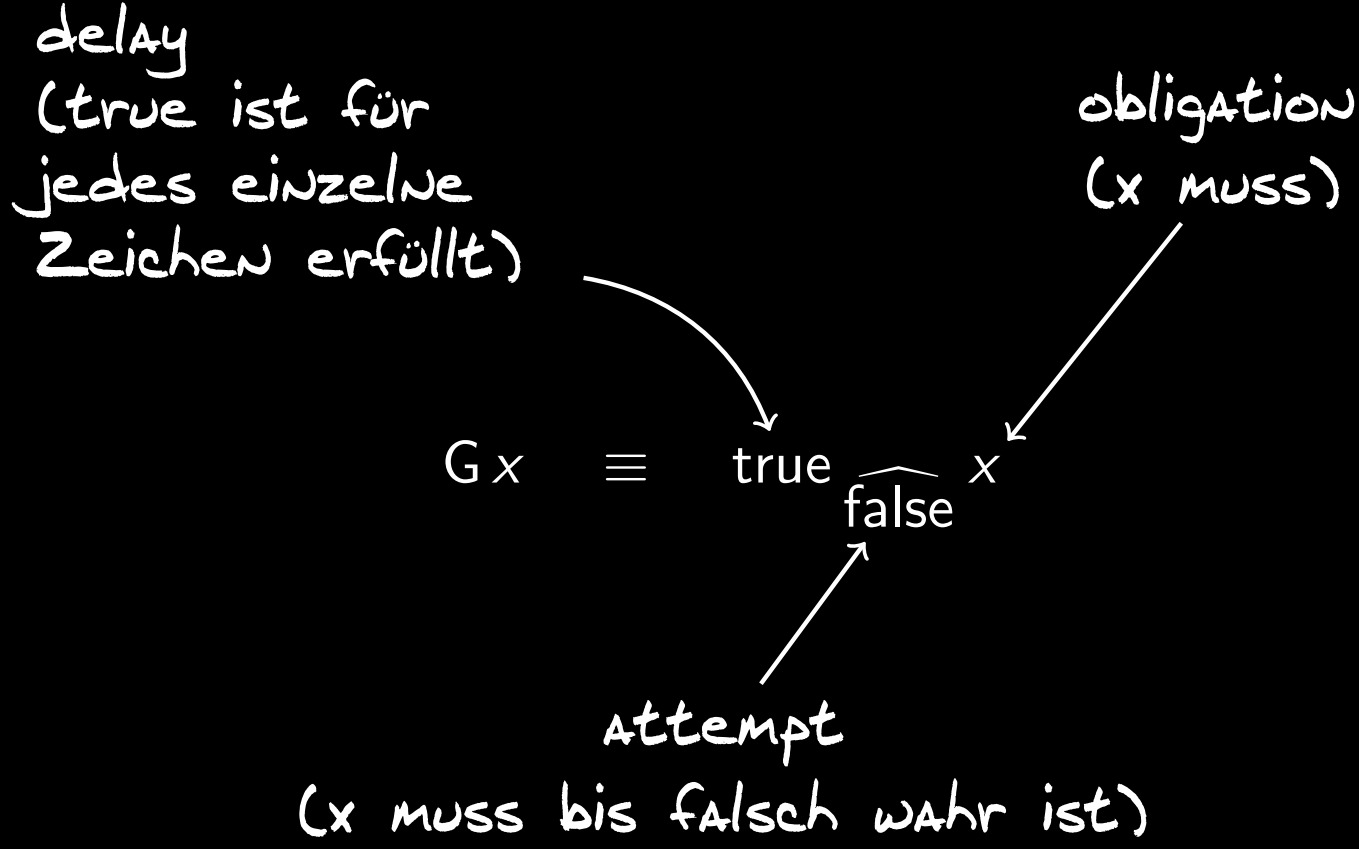
delay
(true ist für
jedes einzelne
Zeichen erfüllt)

$$\mathsf{G}\,x \quad \equiv \quad \mathsf{true} \overset{\frown}{\underset{\mathsf{false}}{}} x$$

Umwandlung LTL in RLTL



Umwandlung LTL in RLTL

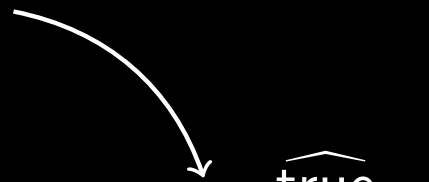


Umwandlung LTL in RLTL

$$\mathsf{F}\, x \quad \equiv \quad \mathsf{true} \,\widehat{\mathsf{true}}\, x$$

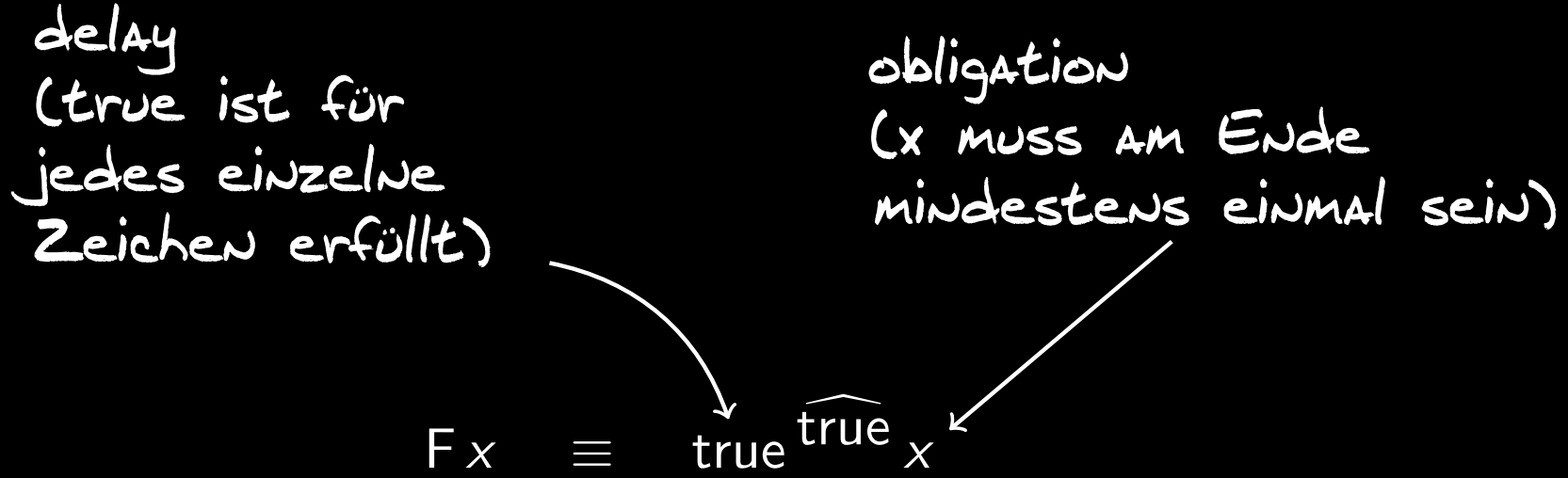
Umwandlung LTL in RLTL

delay
(true ist für
jedes einzelne
Zeichen erfüllt)

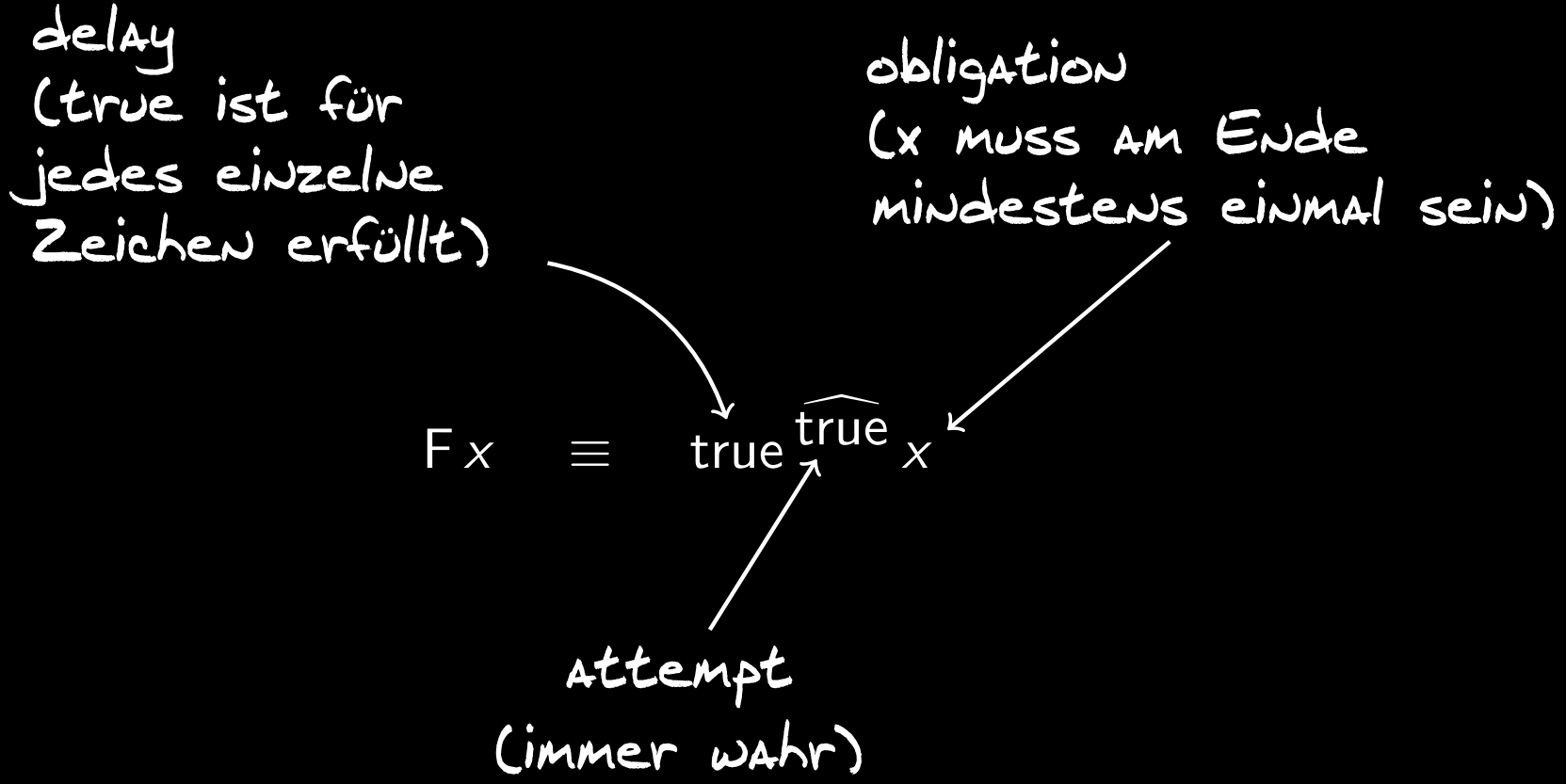


$$\mathsf{F}\,x \quad \equiv \quad \mathsf{true}\,\widehat{\mathsf{true}}\,x$$

Umwandlung LTL in RLTL



Umwandlung LTL in RLTL



Umwandlung LTL in RLTL

$$x \,\mathsf{U}\, y \quad \equiv \quad \mathsf{true}^{\,x}\, y$$

Umwandlung LTL in RLTL

delay
(true ist für
jedes einzelne
Zeichen erfüllt)

$$x \,\mathsf{U}\, y \quad \equiv \quad \mathsf{true}^{\,x}\, y$$

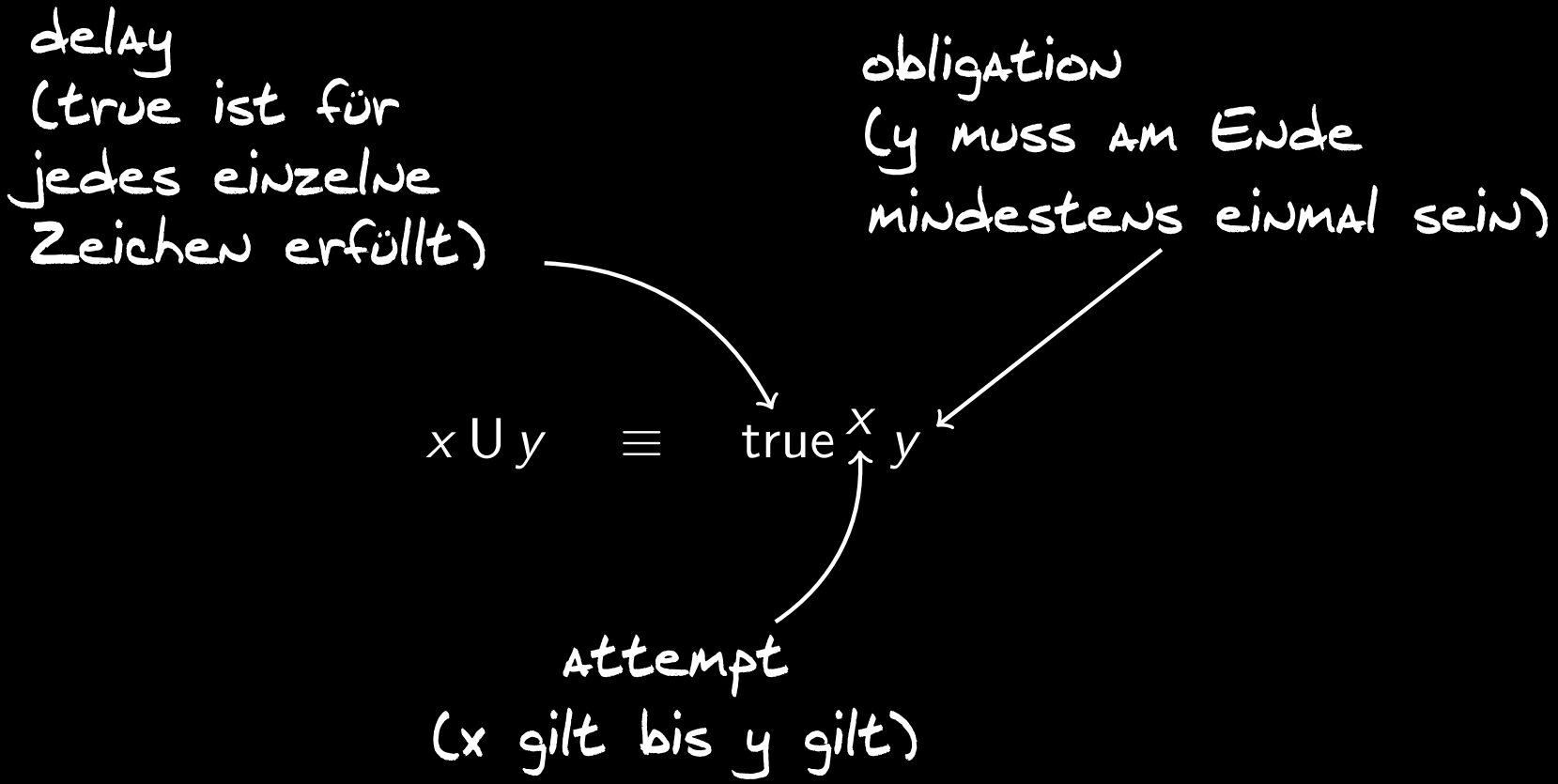
Umwandlung LTL in RLTL

delay
(true ist für
jedes einzelne
Zeichen erfüllt)

obligation
(y muss am Ende
mindestens einmal sein)

$$x \,\mathsf{U}\, y \quad \equiv \quad \mathsf{true}^{\,x}\, y$$

Umwandlung LTL in RLTL



Umwandlung LTL in RLTL

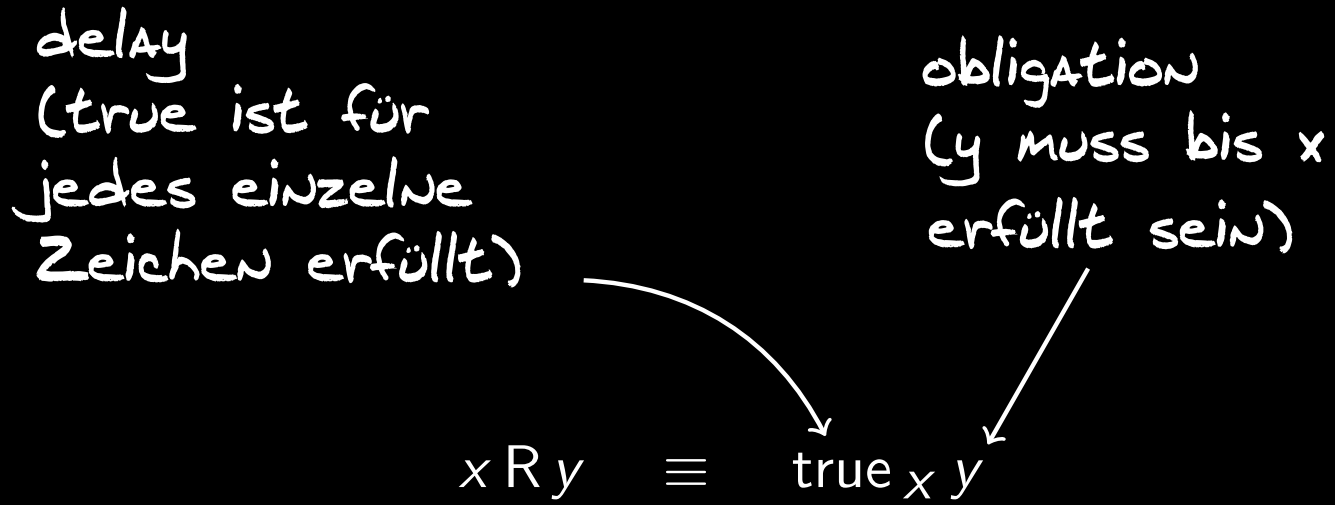
$$x \,\mathsf{R}\, y \quad \equiv \quad \mathsf{true}\, _X\, y$$

Umwandlung LTL in RLTL

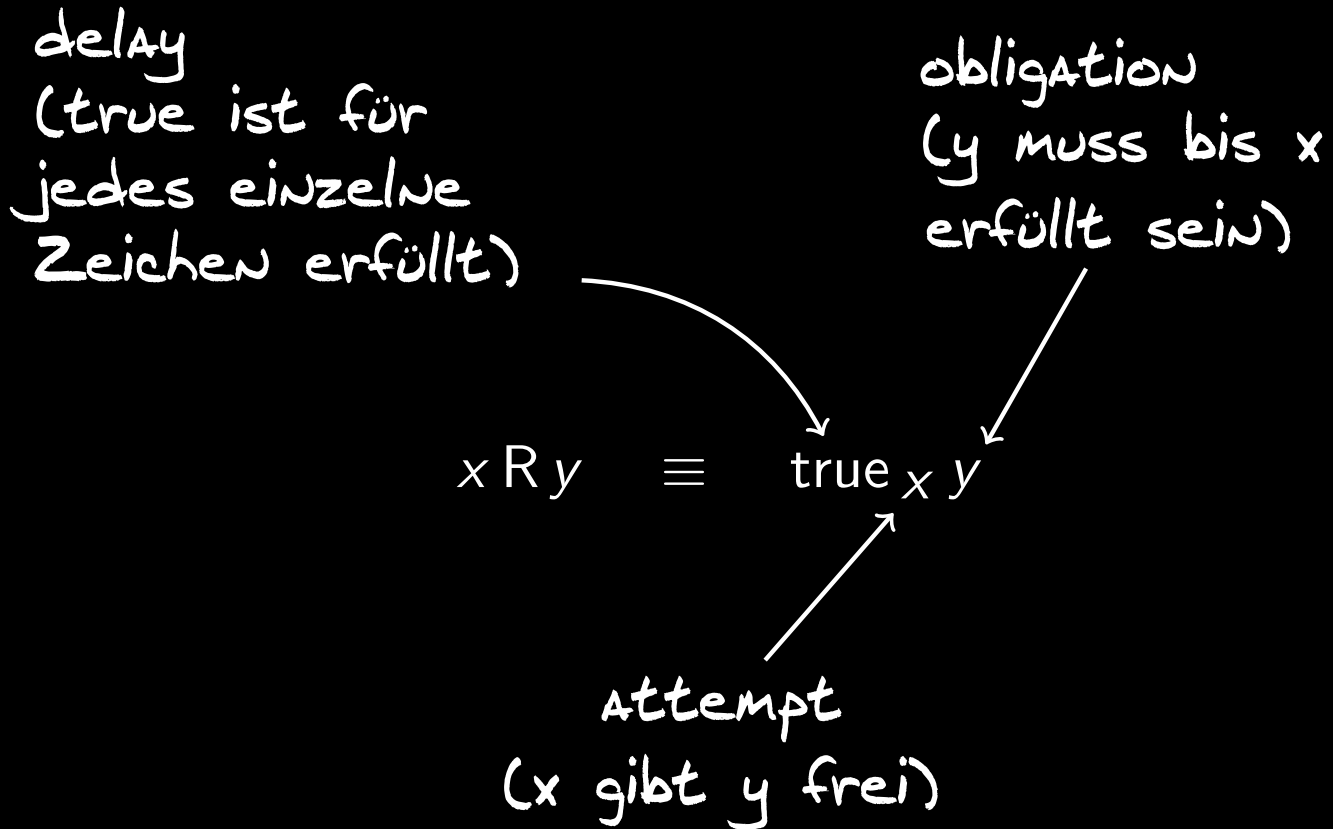
delay
(true ist für
jedes einzelne
Zeichen erfüllt)

$$x \,\mathsf{R}\, y \quad \equiv \quad \mathsf{true}\, _x\, y$$

Umwandlung LTL in RLTL



Umwandlung LTL in RLTL



Zusammenfassung

1. RLTL vereint LTL und ω-reguläre Ausdrücke. Beide können in RLTL überführt werden.
2. RLTL kann alle ω-regulären Sprachen beschreiben.
3. Die Power-Operatoren aus RLTL verallgemeinert die N-, G-, F-, U-, R-Operatoren aus LTL und den ω-Operator aus ω-regulären Ausdrücken.

Anmerkungen von Leucker

- LTL ist ein echtes Subset von ω-regulären Ausdrücken.
- ω-reguläre Ausdrücke können keine Konjunktion und Negation. RLTL und LTL können das. Hier liegt ein wesentlicher Vorteil!
- LTL und ω-reguläre Ausdrücke können linear in RLTL umgewandelt werden. Es findet kein Blow-Up statt.