

# Regel-Spezifikation

---

- [Beschreibung der Regel Notation](#)
- [Regeldefinitionen](#)
- [Qualitative Regeln](#)
  - [OR](#)
  - [AND](#)
  - [yellowOR](#)
  - [redOR](#)
  - [redORyellow](#)
  - [redAND](#)
  - [NONE](#)
- [Quantitative Regeln](#)
  - [GRPxy](#)
  - [yellowGRPxy](#)
  - [redGRPxy](#)
  - [redGRPxyyellow](#)
  - [redNofM](#)
  - [yellowNofM](#)

## Beschreibung der Regel Notation

---

Wir beschreiben die Regeln mit folgender Notation:

```
Farben der Kindknoten -> Farbe der Elternknoten.
```

An die Farben der Kindknoten können Bedingungen an die Häufigkeit gestellt werden. Diese notieren wir mit dem folgendem Schema.

Operatoren	Beispiel	Bedeutung
*	Rot*	Farbe tritt ein oder mehrfach auf
?	Gelb?	Farbe tritt ein oder kein Mal auf
>=xy%	>=xy% Rot	Farbe tritt mindestens xy% auf
<xy%	<xy% Rot	Farbe tritt weniger als xy% auf

Nun sind wir bereit die Regeln klar zu spezifizieren. Die Regeln werden der Reihenfolge nach getestet, bis eine Regel zutrifft.

```
Rot, Rot* -> ___ Rot, (nicht Rot)* -> ___ Gelb, (nicht Rot)* -> ___ [Grün*  
-> Grün]
```

Die letzte Regel ist implizit, da Grün der Normalzustand ist. Wie man überprüft, decken diese Regeln alle Fälle (für nicht-prozentuale Regeln) ab.

Mögliche Werte auf der rechten Seite sind: Rot, Gelb und Grün (auch mit Initialstatus bezeichnet).

## Regeldefinitionen

### Qualitative Regeln

Es folgen alle Regeln, bei denen die Anzahl der Kindknoten nicht relevant ist. Es genügt hierbei also nur die verschiedenen Farben der Kindknoten zu betrachten um die Farbe des Elternknoten zu gewinnen.

#### OR

Farben werden vererbt an den überliegenden Service. Dabei hat die höchste Farbe Vorrang d.h. rot vor gelb, gelb vor grün.

```
Rot, Rot* -> Rot Rot, (nicht Rot)* -> Rot Gelb, (nicht Rot)* -> Gelb
```

Eingabe	Ausgabe
yellow red	red
green green	green
green red	red
green yellow red	red
green yellow	yellow

#### AND

Sämtliche unterliegenden Elemente müssen rot sein, bevor das Resultat ebenfalls rot wird. Sind nicht alle unterliegenden Elemente rot, wird das Resultat gelb. Gelb wird weitervererbt. Rot hat Vorrang vor gelb.

Rot, Rot\* -> Rot Rot, (nicht Rot)\* -> Gelb Gelb, (nicht Rot)\* -> Gelb

Eingabe	Ausgabe
green yellow red	yellow
yellow red	yellow
red red	red
green red	yellow
yellow yellow	yellow

## yellowOR

Rot wird als Gelb vererbt, Gelb als Grün.

Rot, Rot\* -> Gelb Rot, (nicht Rot)\* -> Gelb Gelb, (nicht Rot)\* -> Initialstatus

Eingabe	Ausgabe
red red	yellow
green yellow	green
yellow green red	yellow
yellow red	yellow

Eingabe	Ausgabe
green green	green

## redOR

Rot wird weitervererbt. Gelb wird nicht berücksichtigt und nicht weitervererbt.

Rot, Rot\* -> Rot Rot, (nicht Rot)\* -> Rot Gelb, (nicht Rot)\* -> Initialstatus

Eingabe	Ausgabe
green green	green
green yellow	green
red red	red
yellow red	red
yellow green red	red

## redORyellow

Ergebnis Rot wird gem. Impact Rule nur als gelb weitervererbt.

Rot, Rot\* -> Gelb Rot, (nicht Rot)\* -> Gelb Gelb, (nicht Rot)\* -> Gelb

Eingabe	Ausgabe
red red	yellow

Eingabe	Ausgabe
yellow red	yellow
green green	green
green yellow	yellow
yellow green red	yellow

## redAND

Sämtliche unterliegenden Elemente müssen rot sein, bevor das Resultat ebenfalls rot wird. Gelb wird nicht berücksichtigt und auch nicht weitervererbt.

```
Rot, Rot* -> Rot Rot, (nicht Rot)* -> Initialstatus Gelb, (nicht Rot)* -> Initialstatus
```

Eingabe	Ausgabe
red red	red
yellow green red	green
green green	green
green yellow	green
yellow red	green

## NONE

Es werden keine Farben weitervererbt. Das Resultat ist immer grün. Z.B. Messung von Tätigkeiten, die auf die nächste Ebene keinen direkten Einfluss haben.

Rot\*?, Gelb\*?, Grün\*? -> Initialstatus

Eingabe	Ausgabe
green green	green
red red	green
yellow red	green
yellow green red	green
green yellow	green

## Quantitative Regeln

Es folgen alle Regeln, bei denen die Anzahl der Kindknoten relevant ist.

### GRPxy

Mehr als xy% der unterliegenden Element müssen rot sein, bevor das Resultat rot wird. Ist der geforderte Prozentsatz nicht erreicht, wird das Resultat gelb. Gelb wird weitervererbt. Rot hat Vorrang vor gelb. Mögliche Werte von xy=10, 25, 33, 50, 66, 75

>=xy% Rot\*, (nicht Rot)\*? -> Rot <xy% Rot\*, (nicht Rot)\* -> Gelb Gelb,  
(nicht Rot)\* -> Gelb

Eingabe	Eingabe	Ausgabe
green yellow	0.5	yellow
red red	0.5	red

Eingabe	Eingabe	Ausgabe
yellow red	0.5	red
green green	0.5	green
yellow green red	0.5	yellow

## yellowGRPxy

Mehr als xy% der unterliegenden Element müssen rot sein, bevor das Resultat als gelb weitervererbt wird. Ist der geforderte Prozentsatz nicht erreicht, wird das Resultat grün. Gelb wird weitervererbt. Mögliche Werte von xy=10, 25, 33, 50, 66, 75

```
>=xy% Rot*, (nicht Rot)*? -> Gelb <xy% Rot*, (nicht Rot)* -> Initialstatus  
Gelb, (nicht Rot)* -> Initialstatus
```

Eingabe	Eingabe	Ausgabe
green green	0.5	green
yellow green red	0.5	green
red red	0.5	yellow
green yellow	0.5	green
yellow red	0.5	yellow

## redGRPxy

Mehr als xy% der unterliegenden Element müssen rot sein, bevor das Resultat rot wird. Ist der geforderte Prozentsatz nicht erreicht, wird das Resultat gelb. Gelb wird nicht berücksichtigt und auch nicht weitervererbt. Mögliche Werte von xy=10, 25, 33, 50, 66, 75

```
>=xy% Rot*, (nicht Rot)*? -> Rot <xy% Rot*, (nicht Rot)* -> Gelb Gelb,
(nicht Rot)* -> Initialstatus
```

Eingabe	Eingabe	Ausgabe
yellow green red	0.5	yellow
green yellow	0.5	green
green green	0.5	green
red red	0.5	red
yellow red	0.5	red

## redGRPxyyellow

Mehr als xy% der unterliegenden Elemente müssen rot sein, bevor das Resultat gelb wird. Ist der geforderte Prozentsatz nicht erreicht, wird das Resultat grün. Rot wird nicht berücksichtigt und auch nicht weitervererbt. Mögliche Werte von xy=10, 25, 33, 50, 66, 75

```
>=xy% Rot*, (nicht Rot)*? -> Gelb <xy% Rot*, (nicht Rot)* -> Initialstatus
Gelb, (nicht Rot)* -> Initialstatus
```

Eingabe	Eingabe	Ausgabe
green green	0.5	green
yellow green red	0.5	green
green yellow	0.5	green



Eingabe	Eingabe	Ausgabe
yellow red	0.5	yellow
red red	0.5	yellow

## redNofM

n von m unterliegenden Element müssen rot sein, bevor das Resultat rot wird. Ist der geforderte Prozentsatz nicht erreicht, wird das Resultat gelb. Gelb wird nicht berücksichtigt

m ist nach Vereinbarung stets die Anzahl der Kindelemente. Die Regel stimmt daher überein mit redGRPxy für  $xy=100*n/m$ .

```
>=n/m% Rot*, (nicht Rot)*? -> Rot <n/m% Rot*, (nicht Rot)* -> Gelb Gelb,
(nicht Rot)* -> Initialstatus
```

## yellowNofM

n von m unterliegenden Element müssen rot sein, bevor das Resultat gelb wird. Ist der geforderte Prozentsatz nicht erreicht, wird das Resultat grün. Gelb wird nicht berücksichtigt.

m ist nach Vereinbarung stets die Anzahl der Kindelemente. Die Regel stimmt daher überein mit yellowGRPxy für  $xy=100*n/m$ .

```
>=n/m% Rot*, (nicht Rot)*? -> Gelb <n/m% Rot*, (nicht Rot)* ->
Initialstatus Gelb, (nicht Rot)* -> Initialstatus
```