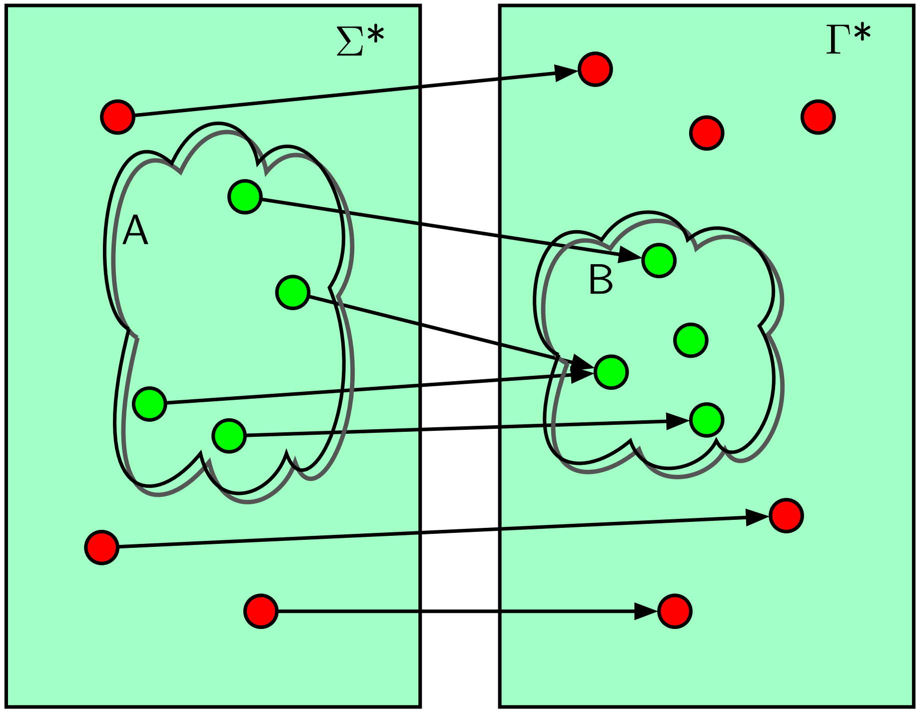
Reduktion

- Seien $A \subset \Sigma^*$ und $B \subset \Gamma^*$.
- Man sagt „A ist **reduzierbar** auf B“ ($A \leq B$) gdw.

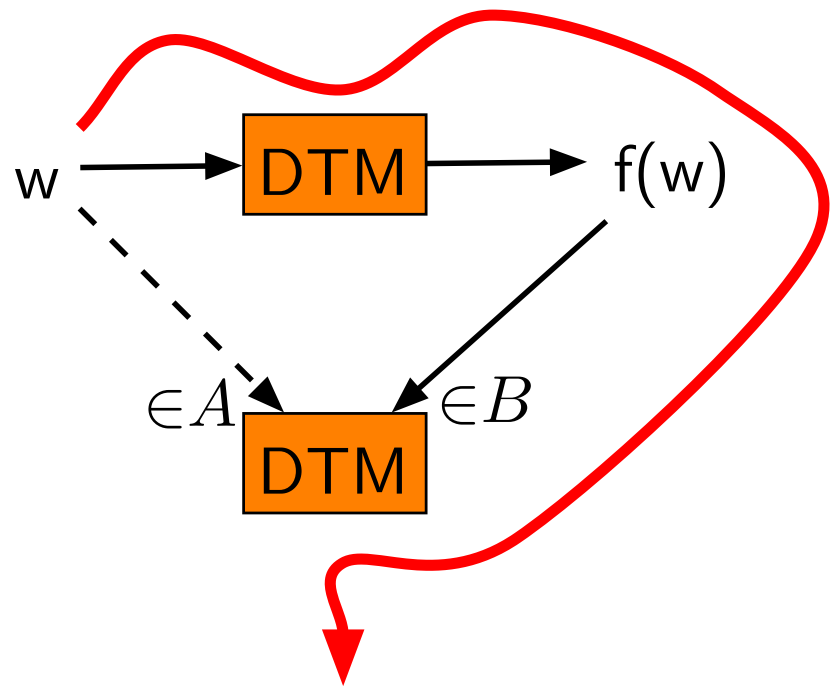
$$\exists f : \Sigma^* \to \Gamma^*. \forall x \in \Sigma^*. x \in A \iff f(x) \in B$$



von speziellem Interesse: Polynomialzeitreduktion ($\leq_{\mathrm{pol}}$), logarithmische-Platz-Reduktion ($\leq_{\log}$).

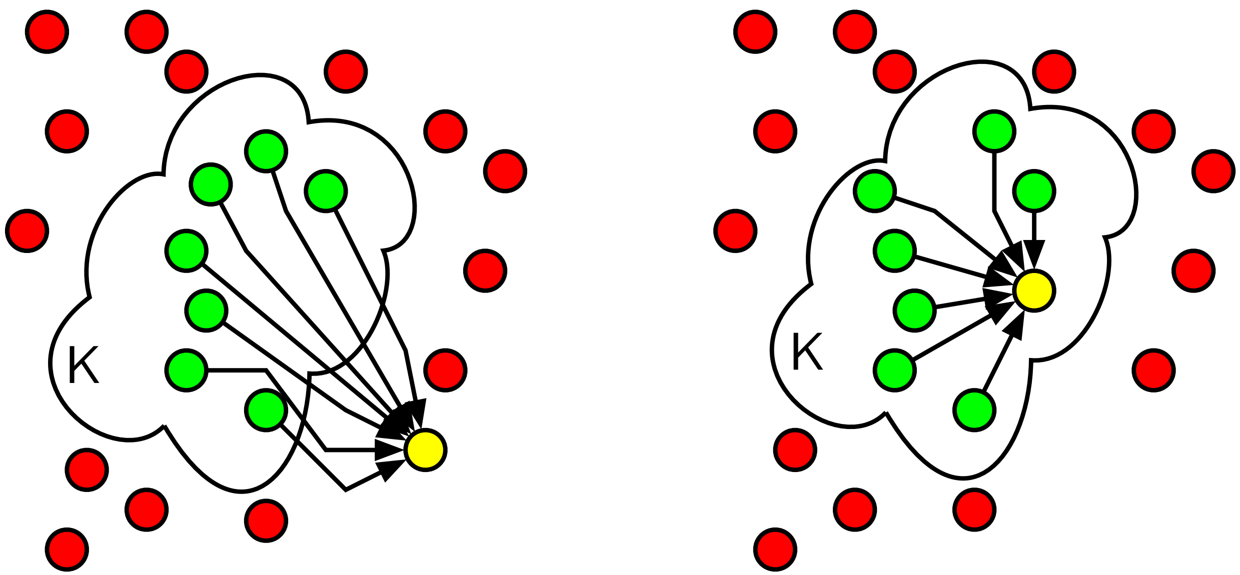
Reduktion (2)

- Zurückführen der Entscheidbarkeit von A auf die Entscheidbarkeit von B:
- Voraussetzung $A \leq B$.



$\mathcal{K}$-Vollständigkeit

- Eine Menge $A \subseteq \Sigma^*$ heißt **hart** (oder besser: **schwer**) für eine Klasse $\mathcal{K}$ gdw. $\quad \forall b \in \mathcal{K}.B \leq A$
- Eine Menge $A \subseteq \Sigma^*$ heißt **vollständig** für eine Klasse $\mathcal{K}$ gdw. $\quad A \in \mathcal{K} \ \wedge \ \forall b \in \mathcal{K}.B \leq A$



$\mathcal{NP}$-Vollständigkeit

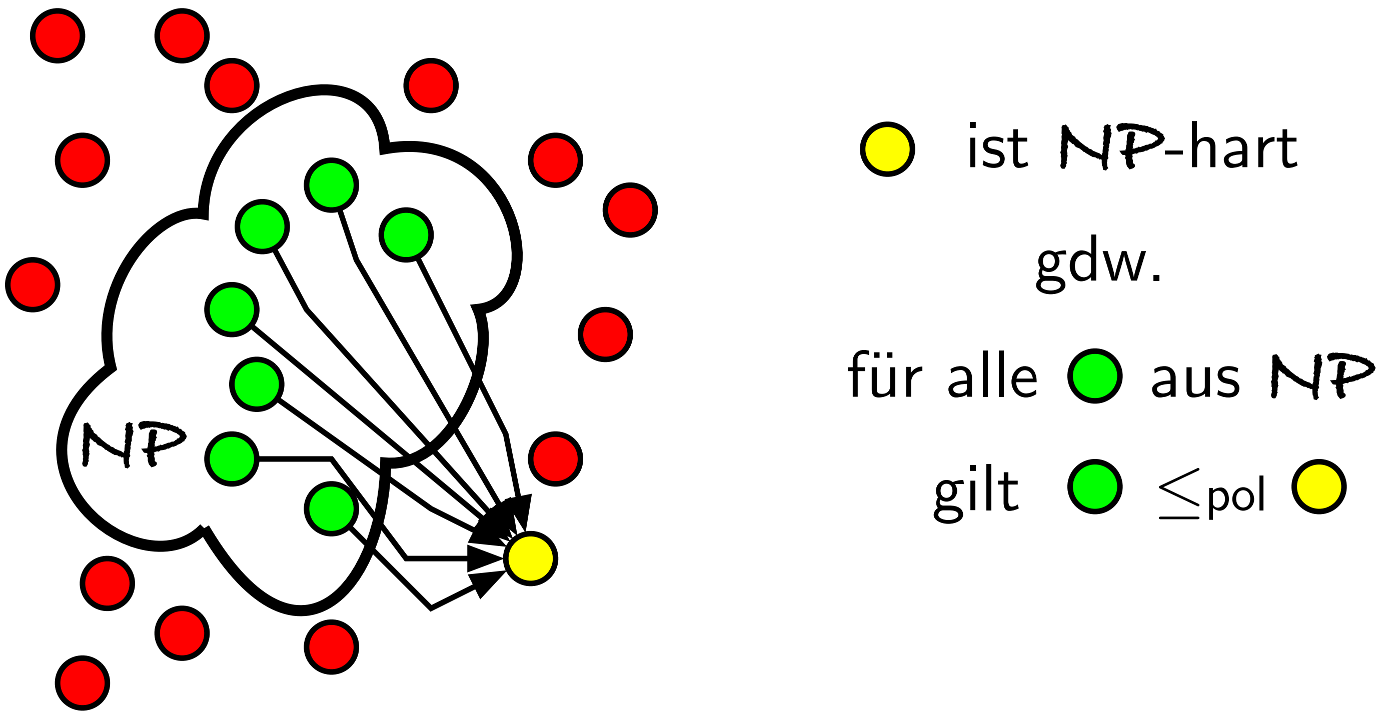
- besonders interessannter Spezialfall der Vollständigkeit:
- Eine Menge $A \subseteq \Sigma^*$ heißt $\mathcal{NP}$**-vollständig** gdw.

$$A \in \mathcal{NP} \ \wedge \ \forall B \in \mathcal{NP}.B \leq_{\mathrm{pol}} A$$

- Ein $\mathcal{NP}$-vollständiges Problem ist somit eines der **schwersten** bzw. **umfassendsten** Probleme innerhalb der Klasse $\mathcal{NP}$.
- Wichtige Eigenschaft: Transitivität von $\leq_{\mathrm{pol}}$.
- Ist A ein $\mathcal{NP}$-vollständiges Problem und gilt $A \leq_{\mathrm{pol}} B$, so ist auch B $\mathcal{NP}$-vollständig.

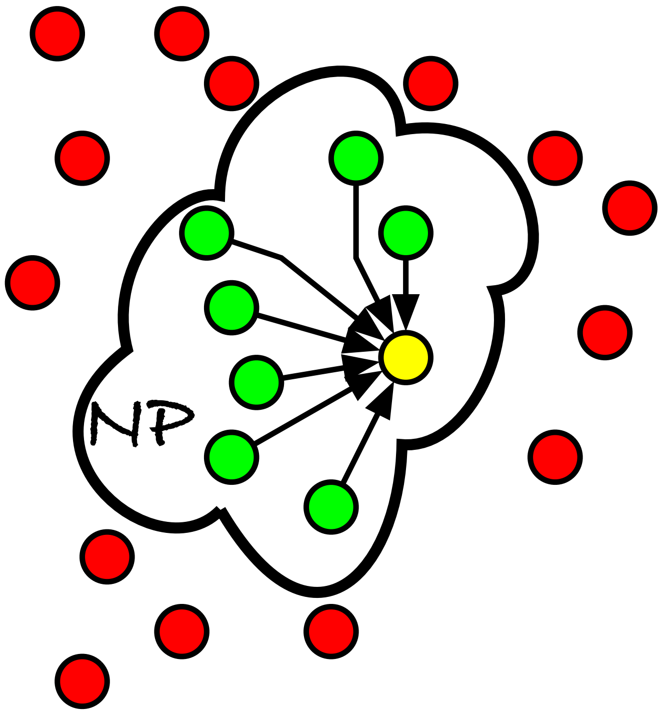
$\mathcal{NP}$-hart

- Eine Menge $A \subseteq \Sigma^*$ heißt $\mathcal{NP}$-**hart** gdw.
  $\forall b \in \mathcal{NP}. B \leq_{\text{pol}} A$



$\mathcal{NP}$-vollständig

- Eine Menge $A \subseteq \Sigma^*$ heißt $\mathcal{NP}$-**vollständig** gdw.
  $A \in \mathcal{NP} \;\wedge\; \forall b \in \mathcal{NP}.B \leq_{\text{pol}} A$



ist NP-vollständig

gdw.

1. liegt in NP

und

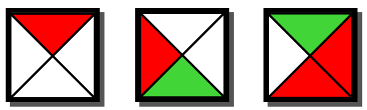
2. ist NP-hart

Ein $\mathcal{NP}$-vollständiges Problem

- Bisher nur Definition der $\mathcal{NP}$-Vollständigkeit
- Aber: Gibt es überhaupt solche Probleme?
  - Diese Frage war lange Zeit ungeklärt!
  - Nach dem ersten folgten aber sofort viele $\mathcal{NP}$-vollständige Probleme.
  - Warum?!
- Idee für ein erstes $\mathcal{NP}$-vollständiges Problem:
  - Codierung aller Polynomialzeit-TM-Rechnungen
  - als Formel (SAT)
  - als Kachelproblem (2D-Domino)

Das Kachelproblem ist in $\mathcal{NP}$

Gegeben:	Eine Menge von r Kacheltypen $\mathcal{R} = \{K_1, K_2, \ldots, K_r\}$, $n \in I\!N$ (beliebig viele Kacheln von jedem Typ)
Gesucht:	Kann man eine Fläche der Größe $n \times n$ korrekt kacheln?
Antwort:	JA oder NEIN

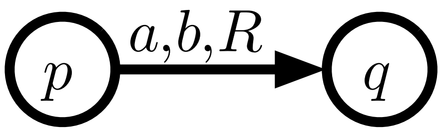


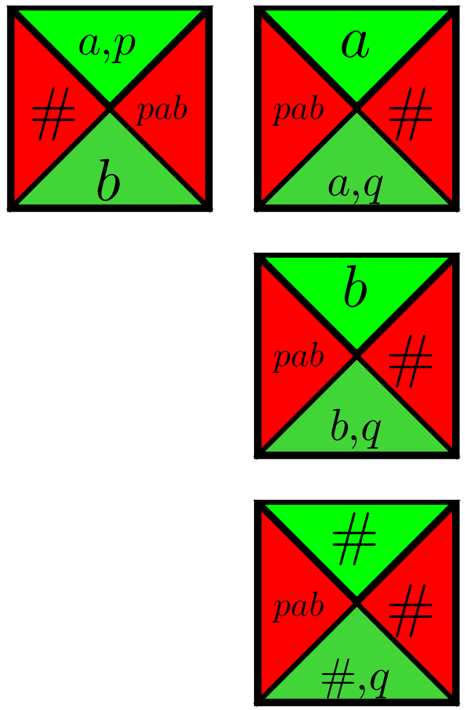
- nur gleichfarbige Seiten dürfen aneinandergelegt werden!
- Steine dürfen nicht gedreht werden!

Das spezielle Kachelproblem

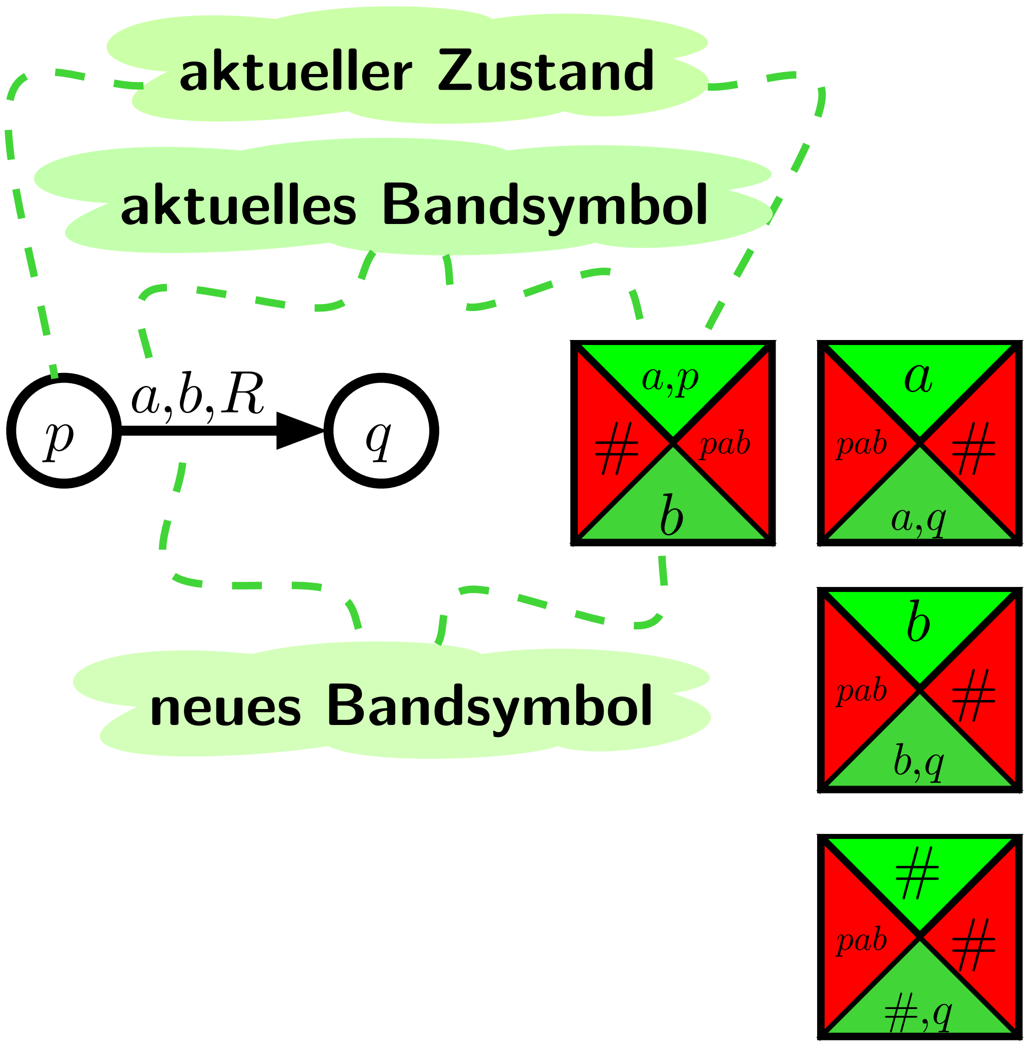
- endlicher Satz von Kacheln
- endliches Spielfeld der Größe $n \times n$
- erste Zeile festgelegt
  - hier: duch die Anfangskonfiguration einer (polynomialzeitbeschränkten) TM
  - Zeilenwechsel entspricht Konfigurationswechsel der TM
- Dieser Spezialfall des Kachelproblems wird nun als $\mathcal{NP}$-vollständig nachgewiesen.

Das spezielle Kachelproblem

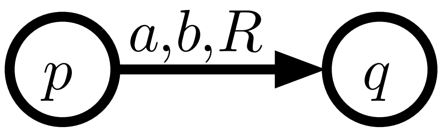


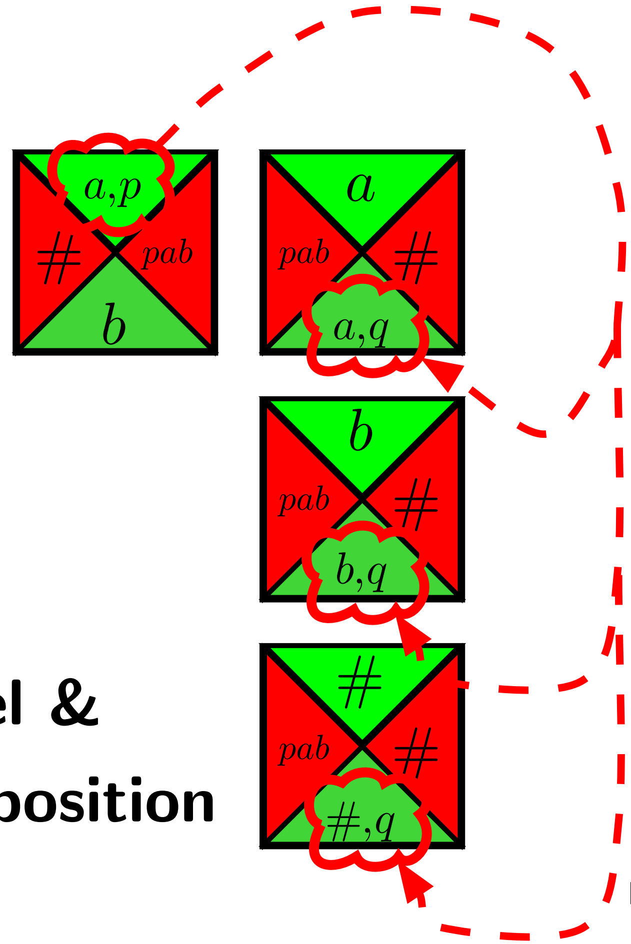


Das spezielle Kachelproblem



Das spezielle Kachelproblem

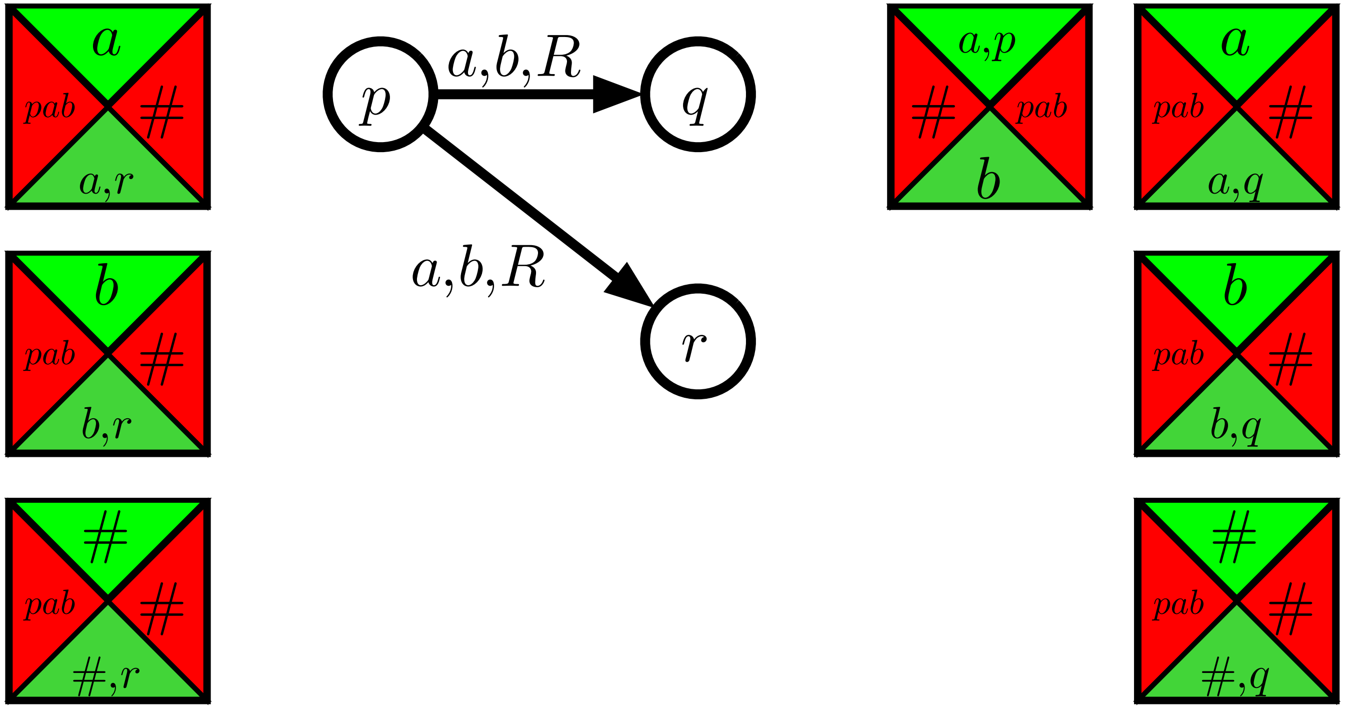




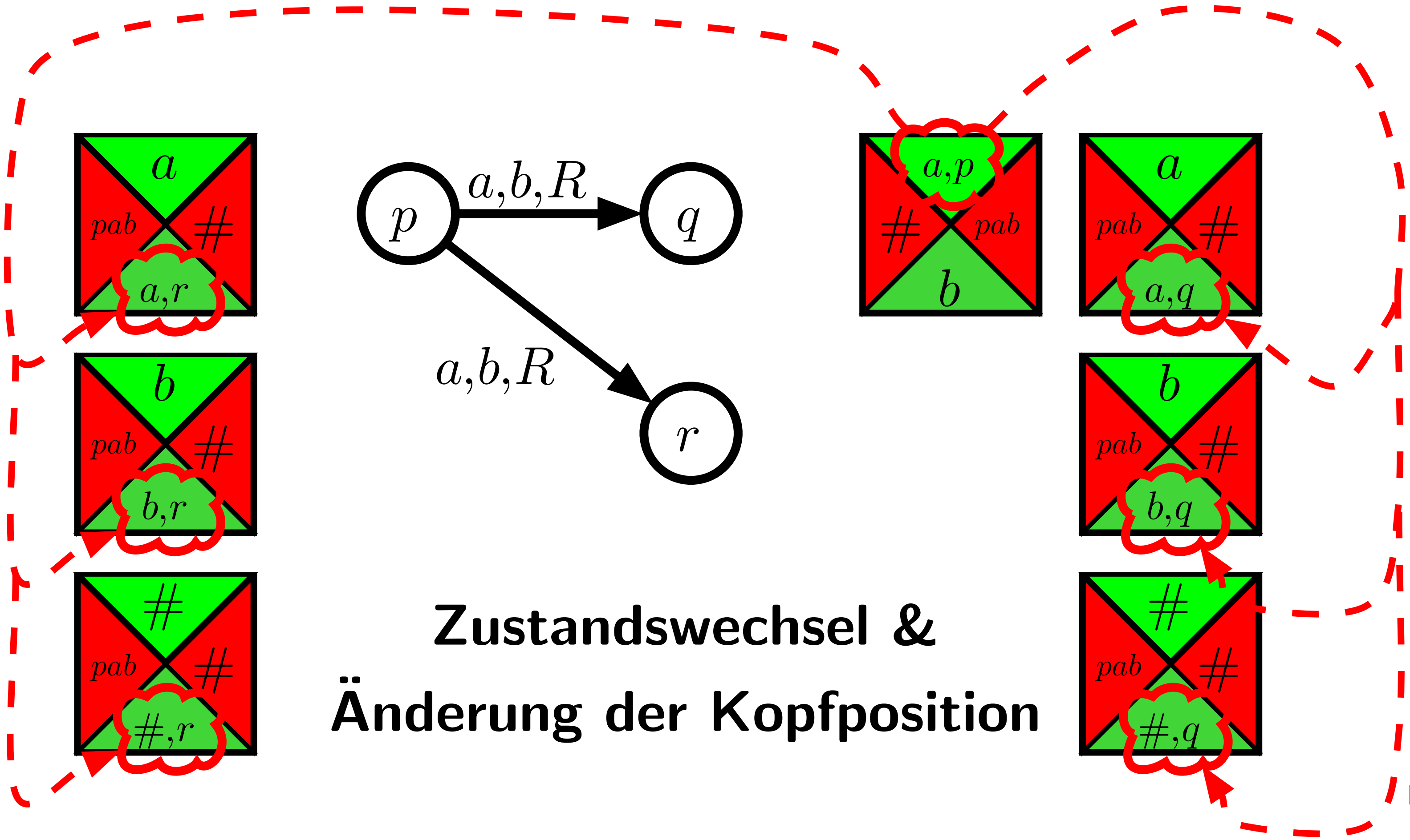
Zustandswechsel &

Änderung der Kopfposition

Das spezielle Kachelproblem

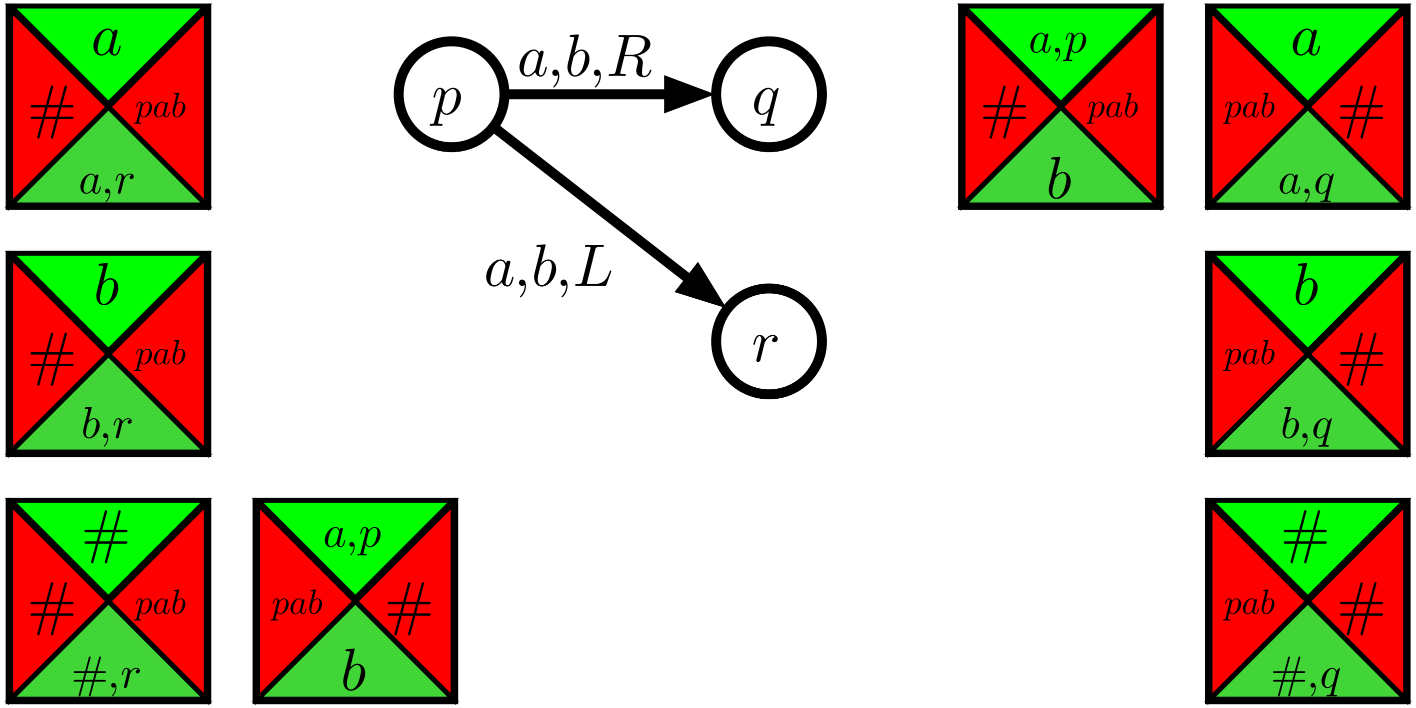


Das spezielle Kachelproblem



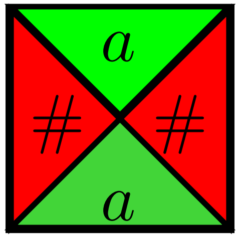
Zustandswechsel &
Änderung der Kopfposition

Das spezielle Kachelproblem



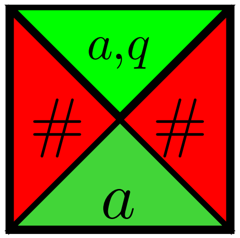
Formale Transformation

Für alle $a \in \Gamma$:



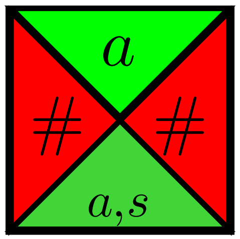
Für alle $q \in Z_{\mathrm{end}}$

und $a \in \Gamma$:

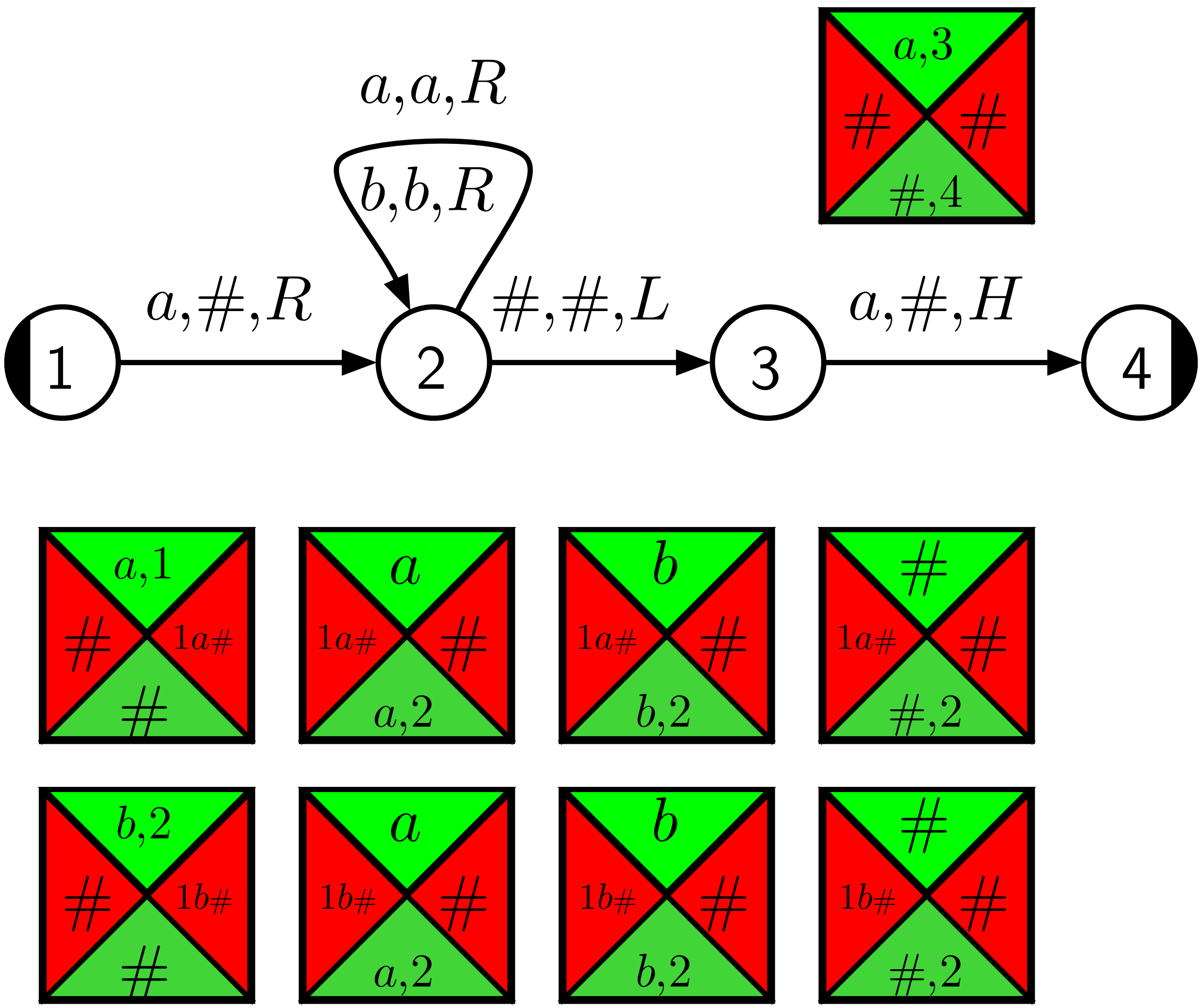


Für Startzustand

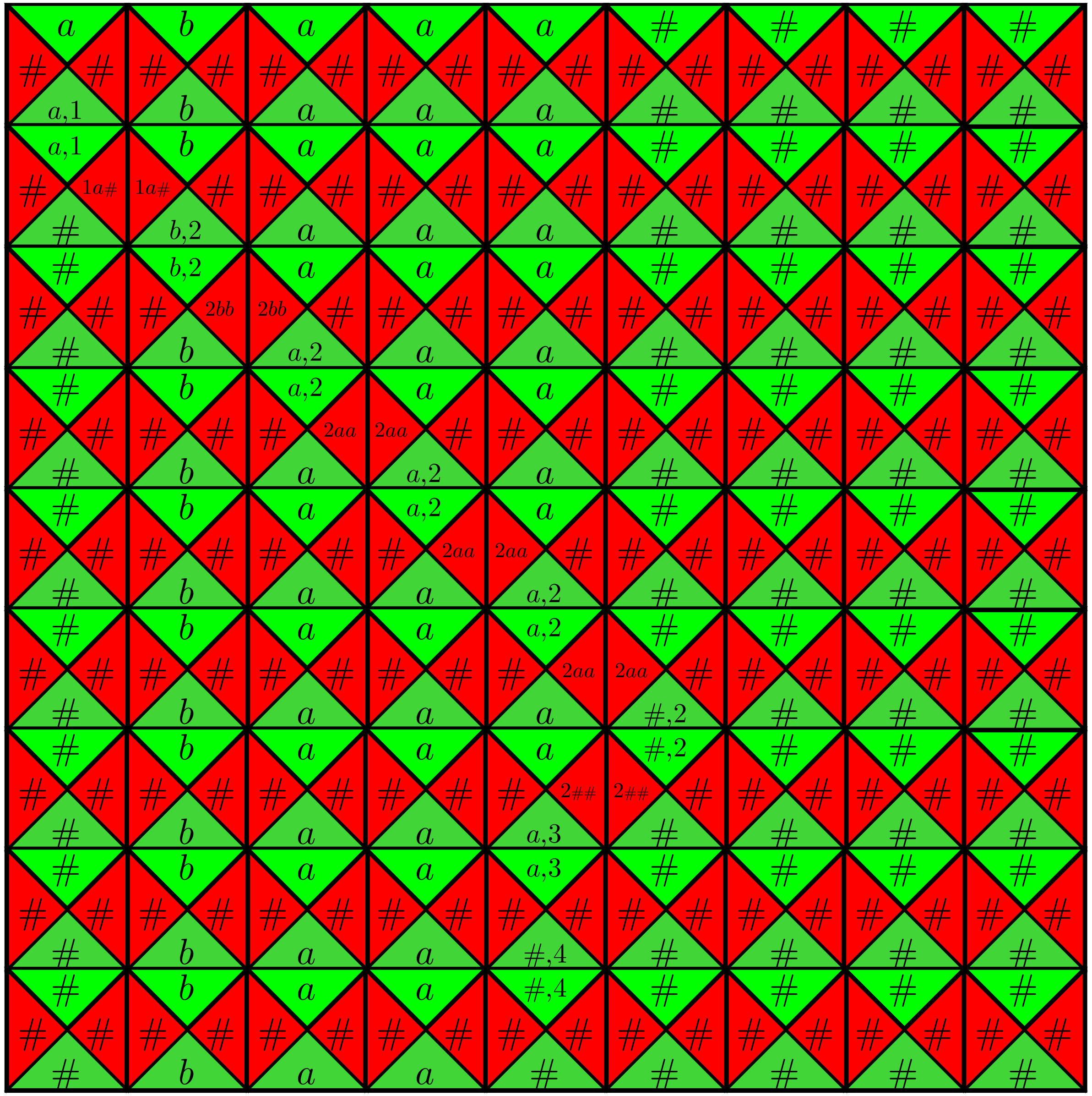
s und alle $a \in \Gamma$:



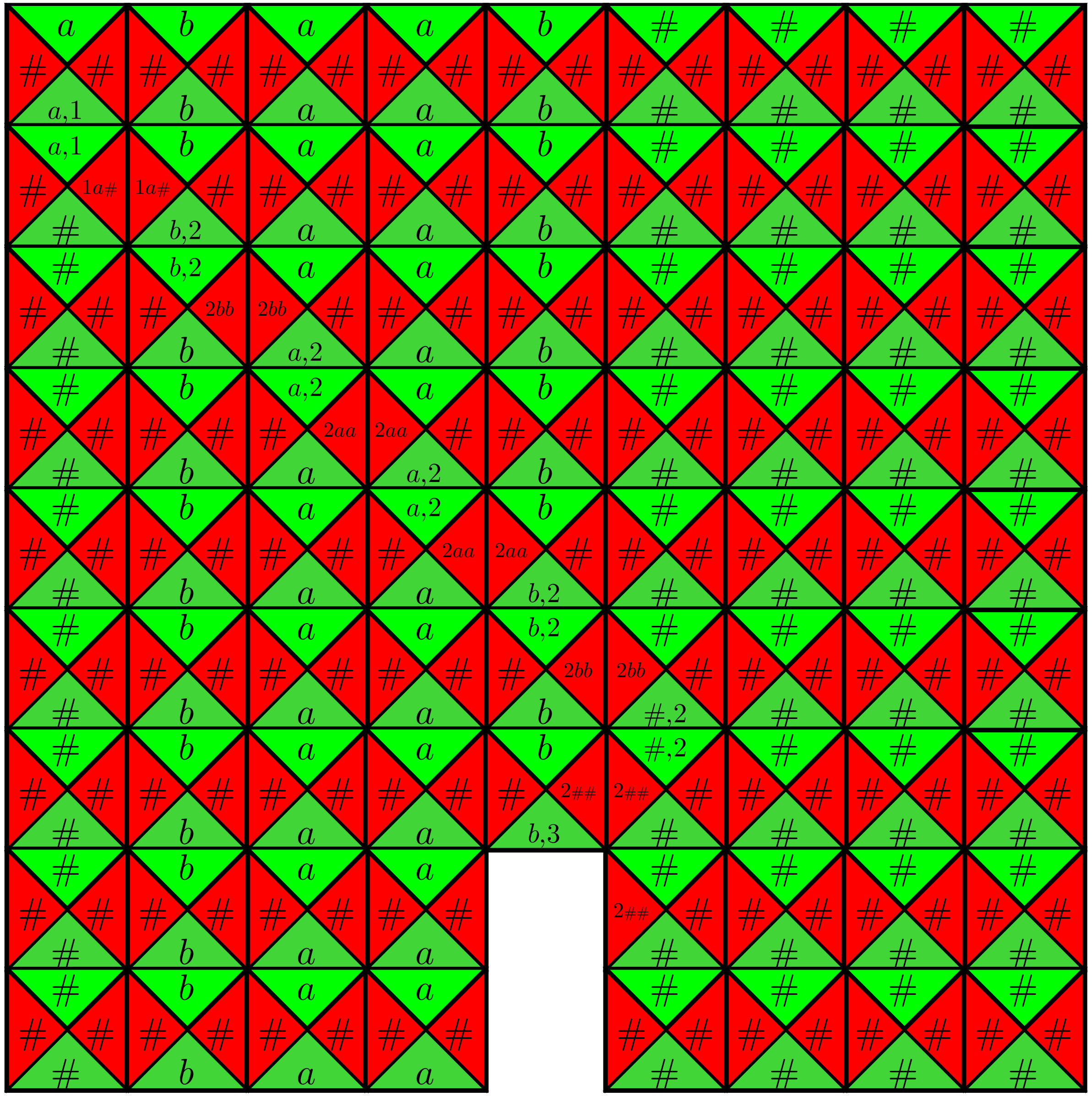
Eine TM M mit $L(M) \in \mathcal{P}$



Beispiel-TM (2)



Beispiel-TM (3)



SAT

Definition: $SAT := \{w \in X^* \mid w$ ist ein erfüllbarer boolescher Ausdruck $\}$ ist das **Erfüllbarkeitsproblem**:

Gegeben:	Eine Menge V von Variablen und eine boolesche (aussagen-logische) Formel $B \in X^*$ mit $X = \{x, 0, 1, \vee, \wedge, \neg, (,)\}$.
Gesucht:	Gibt es eine Belegung der Variablen mit TRUE (1) und FALSE (0), so daß B zu TRUE (1) evaluiert?
Antwort:	JA oder NEIN

$\mathcal{NP}$-Vollständigkeit von SAT

Theorem: SAT ist $\mathcal{NP}$-vollständig.

Beweisidee:

- Es ist einfach zu sehen, daß $SAT \in \mathcal{NP}$ gilt.
  - Zu einer gegebenen Formel B mit den Variablen $x_1, x_2, \ldots x_n$ wird in Linearzeit nichtdeterministisch eine Belegung der Variablen mit TRUE oder FALSE geraten.
  - Es wird geprüft, ob B mit dieser Belegung den Wert TRUE bekommt. Dies ist in Polynomzeit möglich.
- Bleibt zu Zeigen, daß jedes andere Problem aus $\mathcal{NP}$ polynomzeitreduzierbar ist auf SAT.

Reduktion auf SAT

- FELD$(i, j, t) \hat{=}$ In Konfiguration k_t steht das Zeichen x_j in Feld i.
- ZUSTAND$(r, t) \hat{=}$ In der Konfiguration k_t befindet sich die TM A_L im Zustand z_r.
- KOPF$(i, t) \hat{=}$ In der Konfiguration k_t steht der LSK auf dem Feld i.
- Anzahl dieser Variablen in $O(p(n)^2)$, wenn $p(n)$ die Zeitschranke ist. Wir definieren die Formel $\oplus(a_1, a_2, \ldots, a_r) := (a_1 \vee a_2 \vee \ldots \vee a_r) \wedge \bigwedge_{i \neq j}(\neg a_i \vee \neg a_j)$. Die Länge dieser Formel ist von der Ordnung $O(r^2)$.
- Die Formel F_w hat die Form $F_w := A \wedge B \wedge C \wedge D \wedge E \wedge F \wedge G$.

Reduktion auf SAT (2)

Die einzelnen Teilformeln bedeuten:

A $\hat{=}$ In jeder Konfiguration k_t steht der LSK auf genau einem Feld.

B $\hat{=}$ In k_t enthält jedes Feld genau ein Zeichen.

C $\hat{=}$ In k_t befindet sich A_L in genau einem Zustand.

D $\hat{=}$ Bei jedem Übergang wird genau das Feld verändert, auf das der LSK zeigt.

E $\hat{=}$ Jeder Übergang entspricht der Turing-Tafel.

F $\hat{=}$ Die erste Konfiguration ist $k_0 = q_0 w \# \ldots \#$.

G $\hat{=}$ Der Zustand in der letzten Konfiguration $k_{p(n)}$ ist Endzustand aus Z_{end}.

Beispiel einer Teilformel

- Es ist $A := A_1 \wedge A_2 \wedge \ldots \wedge A_{p(n)}$ und
  $\forall t \leq p(n). A_t := \oplus(\text{KOPF}(1,t), \text{KOPF}(2,t), \ldots, \text{KOPF}(p(n),t))$
- Auch B und C sind zusammengesetzte Formeln:

$$
\begin{aligned}
B &:= \bigwedge_{1\leq i,t \leq p(n)} B(i,t) \text{ mit} \\
B(i,t) &:= \oplus(\text{FELD}(i,1,t), \ldots, \text{FELD}(i,m,t)), \; m := |Y| \\
C &:= \bigwedge_{1\leq t \leq p(n)} C_t \text{ mit} \\
C_t &:= \oplus(\text{ZUSTAND}(1,t), \ldots, \text{ZUSTAND}(s,t)), \\
& \quad s := |Z|
\end{aligned}
$$

Einfache Folgerungen

Lemma: Sei L $\mathcal{NP}$-vollständig, $L \leq_{pol} M$ und $M \in \mathcal{NP}$. Dann ist auch M $\mathcal{NP}$-vollständig

Beweis: Das Lemma folgt direkt aus der Definition von $\mathcal{NP}$-Vollständigkeit und der Transitivität der Polynomzeitreduktionen.

- Beispiel im Skript: KNF ist $\mathcal{NP}$-vollständig.
- $KNF := \{w \in X^* \mid w$ ist eine erfüllbare boolesche Formel in konjunktiver Normalform$\}$

Weitere $\mathcal{NP}$-vollständige Probleme

Definition: Das Erfüllbarkeitsproblem boolescher Formeln in konjunktiver Normalform dargestellt als Sprache:
$KNF := \{w \in X^* \mid w$ ist eine erfüllbare boolesche Formel in konjunktiver Normalform$\}$

Theorem: KNF ist $\mathcal{NP}$-vollständig.

Definition: Die Sprache 3-$SAT \subsetneq KNF \subsetneq SAT$ ist gegeben durch:
3-$SAT := \{w \in X^* \mid w$ ist erfüllbare boolesche Formel in konjunktiver Normalform mit genau 3 Literalen in jeder Klausel $\}$.

Theorem: 3-SAT ist $\mathcal{NP}$-vollständig.

$\mathcal{NP}$-vollständige Probleme (2)

Theorem:

- Das Hamilton-Kreis Problem H_c ist $\mathcal{NP}$-vollständig.
- Das Problem L_w („Längster Weg zwischen zwei Knoten“) ist NP-vollständig.
- Das Rucksackproblem ist $\mathcal{NP}$-vollständig.
- Das Partitionierungsproblem von Graphen ist $\mathcal{NP}$-vollständig.
- Das Problem $CLIQUE$ ist $\mathcal{NP}$-vollständig.

$\mathcal{NP}$-vollständige Probleme in $\mathcal{P}$?

- Daß $\mathcal{P} \subseteq \mathcal{NP}$ gilt ist klar!
- Könnte auch das Gegenteil, also $\mathcal{P} \subseteq \mathcal{NP}$, gelten?
  - Man bräuchte nur von einem einzigen $\mathcal{NP}$-vollständigen Problem A zeigen, daß es in $\mathcal{P}$ liegt!
  - Dann würde nämlich sofort folgen:

$$\forall B \in \mathcal{NP} . B \leq_{\text{pol}} A \quad \Rightarrow \quad \forall B \in \mathcal{NP} . B \in \mathcal{P}$$

- Dies ist bislang noch niemandem gelungen!!!