Automaten, Logiken und Spiele

private Mitschrift – *kein* offizielles Script – www.kuertz.net/uni

`Id: AutomatenLogikenSpiele.tex,v 1.41 2006/02/24 13:29:16 mtu Exp`

Disclaimer: Dies ist kein offizielles Script, sondern nur eine private Mitschrift. Ich kann daher keine Gewähr für die Richtigkeit übernehmen. Vor allem weicht die Nummerierung der Kapitel und Sätze zum Teil von der in der Vorlesung verwendeten ab. Falls jemand einen Fehler entdeckt, so möge er/sie mir bitte eine eMail schicken - vielen Dank!

Max Tuengerthal (max@17q.de)

Inhaltsverzeichnis

0 Grundlegende Definitionen und Schreibweisen

Ein *Alphabet* ist eine endliche Menge (Bezeichnung mit A, A', $\ldots$, A_0, $\ldots$). Ein *Buchstabe* ist ein Element eines Alphabetes (Bezeichnung mit a, b, $\ldots$, a', $\ldots$, a_0, $\ldots$). Ein *endliches Wort* is eine Abbildung $\{0, \ldots, n-1\} \to A$ (Bezeichnung mit u, v, $\ldots$). $\{0, \ldots, n-1\}$ sind die Positionen. Mit $u(i)$ wird der Buchstabe in u an Position i bezeichnet, der i-te Buchstabe. Das leere Wort ist ε $(= \emptyset)$. Die *Länge* eines Wortes $u\colon \{0, \ldots, n-1\} \to A$ ist $n =: |u|$, d.h., $u\colon \{0, \ldots, |u|-1\} \to A$. Seien u, v endliche Wörter über A, d.h., $u, v \in A^*$. Dann ist $u \cdot v\colon \{0, \ldots, |u| + |v| - 1\} \to A$ (Konkatenation),

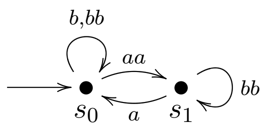
$$u \cdot v(i) = \begin{cases} u(i) & \text{falls} i < |u| \\ v(i - |u|) & \text{falls } |u| \leq i < |u| + |v| \end{cases}$$

A^* is die Menge aller endlichen Wörtern über A. $A^+ = A^* \setminus \{\varepsilon\}$. Eine *(formale) Sprache* ist eine Teilmenge von A^* für ein Alphabet A.

Teil I
Endliche Automaten auf endlichen Wörtern

1 Transitionssysteme (Halbautomaten) und Akzeptierbedingungen

Ein *endliches Transitionssystem (endlicher Halbautomat) mit Worttransitionen* über A ist ein Tupel $\Pi = (A, S, s_I, \Delta)$ mit A Alphabet, S endliche Zustandsmenge, $s_I \in S$ Anfangszustand und $\Delta \subseteq S \times A^* \times S$ *endliche* Transitionsrelation.

Beispiel 1. Das Transitionssystem 

ist formal gegeben durch $(\{a, b\}, \{s_0, s_1\}, s_0, \{(s_0, b, s_0), (s_0, bb, s_0), (s_0, aa, s_1), \ldots\})$.

Ein *Pfad* durch Π ist eine endliche Folge u über $A^* \cup S$ mit folgenden Eingenschaften

- $u(0) = s_I$
- $|u|$ ungerade
- für alle geraden $0 \leq i < |u| - 2$ gilt $(u(i), u(i+1), u(i+2)) \in \Delta$.

Die *Beschriftung* des Pfades ist $\beta(u) = u(1)u(3)\ldots u(|u|-2)$.

Beispiel 2. Ein Pfad in $\mathbb{A}$ aus Beispiel 1 ist $u = s_0, bb, s_0, aa, s_1, a, s_0$. Er hat die Beschriftung $\beta(u) = bbaaa$.

Ein *endlicher Automat* ist ein Tupel $\mathbb{A} = (A, S, s_I, \Delta, F)$ mit (A, S, s_I, Δ) Transitionssystem und $F \subseteq S$ Endzustandsmenge (Menge der akzeptierenden Zustände). Ein Pfad heißt *akzeptierend*, falls $u(|u|-1) \in F$. $\Pi(\mathbb{A})$ ist die Menge der Pfade von $\mathbb{A}$. und $\Pi_{\text{akz}}(\mathbb{A})$ ist die Menge der akzeptierenden Pfade von $\mathbb{A}$.

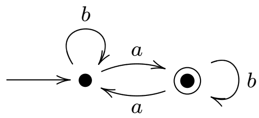
$$L_\exists(\mathbb{A}) = \{v \in A^* \mid \exists u \in \Pi_{\text{akz}}(\mathbb{A}) : \beta(u) = v\}$$

ist die *nicht-deterministisch* (oder *existenziell*) erkannte Sprache.

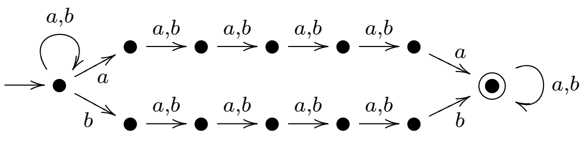
$$L_\forall(\mathbb{A}) = \{v \in A^* \mid \forall u \in \Pi(\mathbb{A}) : (\beta(u) = v \rightarrow u \in \Pi_{\text{akz}}(\mathbb{A}))\}$$

ist die *universell* erkannte Sprache.

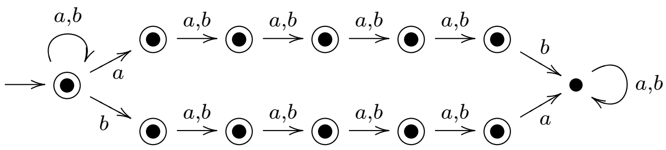
Beispiel 3. (a) $L_\exists(\mathbb{A}) =$ „Anzahl a's ungerade“ und
$L_\forall(\mathbb{A}) =$ „Anzahl a's ungerade“ für

$\mathbb{A} =$ 

(b) $L_\exists(\mathbb{A}) = \{u \mid \exists i : u(i) = u(i+5)\}$ und
$L_\forall(\mathbb{A}) = \emptyset$ für

$\mathbb{A} =$ 

(c) $L_\exists(\mathbb{A}) = \{a, b\}^*$ und
$L_\forall(\mathbb{A}) = \{u \mid \forall i : i + 5 < |u| \rightarrow u(i) = u(i+5)\}$ für

$\mathbb{A} =$ 

2 Alternierende endliche Automaten

Idee. Kombination von existenziellem und universellem Akzeptieren.

Ein Transistionssystem heißt *ε-kritisch*, wenn es eine Folge $s_0, s_1, \dots, s_n$ mit $s_0 = s_n$ und $(s_i, \varepsilon, s_{i+1}) \in \Delta$ für alle $i = 0, \dots, n-1$ gibt.
Ein *alternierender endlicher Automat* ist ein Tupel $\mathbb{A} = (A, S, s_I, \Delta, E, U, F)$, wobei (A, S, s_I, Δ, F) ein endlicher ε-unkritischer Automat ist und U und E den Zustandsraum S partitionieren. U ist die Menge der *universellen Zustände* und E die Menge der *existentiellen Zustände*.

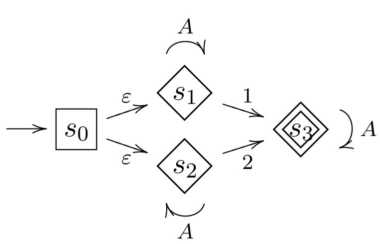
Idee.
- In einem existentiellen Zustand muss sich der Automat einen Nachfolger aussuchen, an dem er weiter rechnet.
- In einem universellen Zustand muss der Automat alle Nachfolger weiter verfolgen.

Notation 4.
- Universelle Zustände werden durch $\Box$ bezeichnet (Verbindung zur Modallogik: notwendigerweise).
- Existentielle Zustände werden durch $\Diamond$ bezeichnet (Verbindung zur Modallogik: möglicherweise).

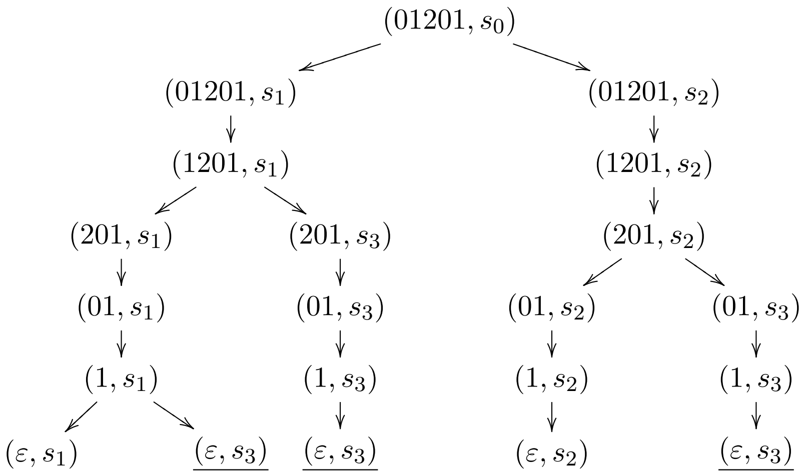
Sei $\Pi = (A, S, s_I, \Delta)$ ein Transitionssystem und $u \in A^*$ ein beliebiges Wort. Der *Berechnungsbaum* $t_{\mathbb{A}}(u)$ von Π auf u ist der kleinste Baum mit folgenden Eigenschaften:

- (u, s_I) ist die Wurzelbeschriftung und
- ist $(s, v, s') \in \Delta$ und ist (vv', s) eine Knotenbeschriftung eines Knoten w, so hat w einen mit (v', s') beschrifteten Nachfolger.

Beispiel 5. Sei $A = \{0, 1, 2\}$ und $L = \{u \in A^* \mid 1 \text{ und } 2 \text{ treten in } u \text{ auf}\}$ und $\mathbb{A}$ wie folgt:



Es gilt $L(\mathbb{A}) = L$. Der alternierender Automat $\mathbb{A}$ verfolgt „gleichzeitig“ beide Buchstaben (1 und 2) unabhängig voneinander.
Der Berechnungsbaum von $\mathbb{A}$ für $u = 01201$ ist folgender:



Ein Wort u wird von $\mathbb{A}$ *akzeptiert*, falls $t_{\mathbb{A}}(u)$ einen Teilbaum t mit folgenden Eigenschaften enthält:

- Die Wurzel von $t_{\mathbb{A}}(u)$ gehört zu t.
- für jeden Knoten w von t mit Beschriftung (v, s) wobei $v \neq \varepsilon$ gilt:
    1. Falls $s \in E$, so besitzt w einen Nachfolger in t.
    2. Falls $s \in U$, so gehören alle Nachfolger von w in $t_{\mathbb{A}}(u)$ auch zu t.
- für jeden Knoten w von t mit Beschriftung (ε, s) gilt:
    1. Falls $s \in E \setminus F$, so besitzt w einen Nachfolger in t.
    2. Falls $s \in U$, so gilt $s \in F$ und alle Nachfolger von w in $t_{\mathbb{A}}(u)$ gehören auch zu t.

Bemerkung 6. Zu $\mathbb{A} = (A, S, s_I, \Delta, F)$ definiere $\mathbb{A}_\exists = (A, S, s_I, \Delta, S, \emptyset, F)$ und $\mathbb{A}_\forall = (A, S, s_I, \Delta, \emptyset, S, F)$. Dann gilt

$$L_\exists(\mathbb{A}) = L(\mathbb{A}_\exists) \quad \text{und} \quad L_\forall(\mathbb{A}) = L(\mathbb{A}_\forall).$$

Ziel. Äquivalenz von alternierenden und deterministisch endlichen Automaten!

Lemma 7 (Schaltkreischarakterisierung)**.** *Sei $\mathbb{A}$ ein alternierender endlicher Automat über A und $u \in A^*$. Sei $t = t_{\mathbb{A}}(u)$ und $s = s_{\mathbb{A}}(u)$ der* Schaltkreis*, der aus t wie folgt ensteht:*

- *Jede Beschriftung* (v, s) *eines Blattes wird ersetzt durch* 1 *genau dann, wenn* ($s \in U$ *und* $v \neq \varepsilon$) *oder* ($s \in F$ *und* $v = \varepsilon$)*; sonst durch* 0 *und*

- *jede Beschriftung* (v, s) *eines inneren Knoten wird ersetzt durch* $\wedge$ *genau dann, wenn* $s \in U$*; sonst durch* $\vee$*. Ausnahme: Wenn* $v = \varepsilon$ *und* $s \in F \cap E$*, dann wird die Beschriftung durch die* 1*-Funktion ersetzt. Wenn* $v = \varepsilon$ *und* $s \in U \setminus F$*, dann wird die Beschriftung durch die* 0*-Funktion ersetzt.*

Dann gilt $u \in L(\mathbb{A})$ *genau dann, wenn* $s_{\mathbb{A}}(u)$ *zu* 1 *ausgewertet wird.*

Beweis. Zu zeigen:

$$s_{\mathbb{A}}(u) \text{ liefert } 1 \Leftrightarrow \text{es ex. ein akz. Teilbaum } t \text{ von } t_{\mathbb{A}}(u)$$

„$\Rightarrow$“ **Idee:** gewinne einen Teilbaum aus dem mit Einsen und Nullen bewerteten Schaltkreis. Gehe dann *top-down* vor: Z. B. bei $\vee$-Gatter: Es muss einen Eingang geben, der mit 1 bewertet ist. Den Knoten nimmt man in den Teilbaum t auf. Z. B. bei inneren Knoten, der mit (ε, s) beschriftet ist: Falls dieser mit 1 bewertet, dann weil

  - $s \in U \cap F$: Dann nehme alle Nachfolger hinzu.
  - $s \in E \cap F$: Dann sind wir bei einem 1-Gatter und schließen den Baum dort ab.
  - $s \in E \setminus F$: Dann sind wir bei einem $\vee$-Gatter und es gibt einen mit 1 bewerteten Nachfolger.

„$\Leftarrow$“ Sei t ein akzeptierender Teilbaum. Wir zeigen nun, dass die Bewertung der Blätter von t mit 1 verträglich mit dem Schaltkreis ist (D. h., ein Blatt entspricht einem 1-Gatter oder einer 1 als Eingabe) und, dass unter dieser potenziellen Bewertung, an den inneren Knoten immer eine 1 als Bewertung auftritt. □

Folgerung 8 (Komplementaritätsprinzip)**.** *Ist* $\mathbb{A} = (A, S, s_I, \Delta, E, U, F)$ *ein* ε*-unkritischer alternierender endlicher Automat, so gilt*

$$\overline{L(\mathbb{A})} = L(\overline{\mathbb{A}})$$

mit $\overline{\mathbb{A}} = (A, S, s_I, \Delta, U, E, S \setminus F)$.

Jeder alternierende endliche Automat (AEA) ist unter Komplement abgeschlossen. Jeder nicht-deterministische Automat ist äquivalent zu einem AEA. Noch zu zeigen: Jeder AEA ist äquivalent zu einem nicht-deterministischen Automaten (NEA). Dies zeigen wir in zwei Schritten:

1. spezieller Fall: $\mathbb{A}$ ist *alphabetisch*, d. h., $\Delta \subseteq S \times A \times S$.
2. allgemeiner Fall

Satz 9. *Jeder AEA ist äquivalent zu einem NEA, d. h. jede von einem AEA erkannte Sprache ist regulär.*

Beweis. Wir beweisen dies in drei Schritten.

1. „alphabetischer AEA $\rightarrow$ NEA“.

   Wir führen eine so genannte *De-Universalisierungskonstruktion* durch, d. h. wir wollen die $\Box$ entfernen!

   Idee: $\Box$ bedeuted, dass mehrere Pfade im Berechnungsbaum verfolgt werden müssen. Benutze einfach als Zustände im neuen NEA Mengen von Zuständen des alten Automaten. Alle Zustände in einer solchen Menge werden weiterverfolgt!

   Aus $\mathbb{A} = (A, S, s_I, \Delta, E, U, F)$ wird

   $$\mathbb{A}_{\mathrm{du}} = (A, 2^S, \{s_I\}, \Delta', 2^F),$$

   wobei $(S', a, S'') \in \Delta'$ gdw. für alle $s \in S'$ gilt:

   - falls $s \in E$, so gibt es $(s, a, s') \in \Delta$ mit $s' \in S''$.
   - falls $s \in U$, so gilt für alle $(s, a, s') \in \Delta$: $s' \in S''$.

   Zeige dann: $L(\mathbb{A}_{\mathrm{du}}) = L(\mathbb{A})$.

   Vorstellung: Der NEA arbeitet den Berechnungsbaum scheibchenweise ab.

2. „AEA mit kurzen Transitionen $\rightarrow$ ε-NEA“ (kurze Transitionen heißt, dass für $(s, u, s') \in \Delta$ gilt, dass $|u| \leq 1$).

   Probleme:

   - Was passiert in Situationen der Art $\begin{tikzcd}[column sep=small, row sep=small] & \Box \arrow[dl, "a"'] \arrow[dr, "\varepsilon"] & \\ {} & & {} \end{tikzcd}$?
   - Wie schaffen wir es, den Automaten nach Lesen des gesamten Wortes dazu zu zwingen, evtl. weitere ε-Transitionen auszuführen? $\begin{tikzcd}[row sep=small] \boxdot \arrow[d, "\varepsilon"] \\ \Diamond \end{tikzcd}$

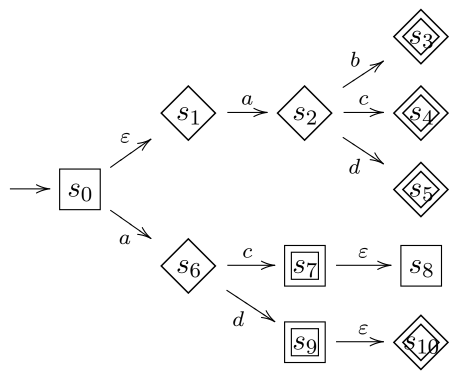
   Ansatz: Anstelle einer Menge von Zuständen S' als neuen Zustand benutzen wir ein Paar (M, N) von Mengen von Zuständen als neuen Zustand. M ist die Menge von Zuständen, von denen aus noch

ε-Transitionen ausgeführt werden müssen. N ist die Menge von Zuständen die schon abgehakte Zustände enthält.

Die neue Zustandsmenge ist $2^S \times 2^S$. Der Anfangszustand ist $(\{s_I\}, \emptyset)$. Die Endzustandsmenge ist $\{(\emptyset, N) \mid N \subseteq F\}$. Die Transitionsrelation ist definiert durch

- $((M_0, N_0), \varepsilon, (M_1, N_1)) \in \Delta'$ genau dann, wenn
  - $N_0 \subseteq N_1$,
  - für alle $s \in M_0 \cap E$ gibt es $s' \in M_1$ mit $(s, \varepsilon, s') \in \Delta$ oder $s \in N_1$ und
  - für alle $s \in M_0 \cap U$ und s' mit $(s, \varepsilon, s') \in \Delta$ gilt $s' \in M_1$ und $s \in N_1$.
- $((M_0, N_0), a, (M_1, N_1)) \in \Delta'$ genau dann, wenn
  - $M_0 = \emptyset$,
  - für alle $s \in N_0 \cap E$ gibt es $s' \in M_1$ mit $(s, a, s') \in \Delta$ und
  - für alle $s \in N_0 \cap U$ und alle $(s, a, s') \in \Delta$ gilt $s' \in M_1$.

Beispiel 10. $\mathbb{A} =$



Es gilt $ab, ac \notin L$ und $ad \in L$. Der Berechnungsbaum für den aus $\mathbb{A}$ konstruierten ε-NEA ist:

$$\begin{tikzcd}
{} \arrow[r] & (\{s_0\}, \emptyset) \arrow[d, "\varepsilon"] & \boxed{\boxed{(\emptyset, \{s_5, s_9, s_{10}\})}} & \boxed{(\emptyset, \{s_4, s_7, s_8\})} \\
& (\{s_1\}, \{s_0\}) \arrow[d, "\varepsilon"] & (\{s_{10}\}, \{s_5, s_9\}) \arrow[u, "\varepsilon"] & (\{s_8\}, \{s_4, s_7\}) \arrow[u, "\varepsilon"] \\
& (\emptyset, \{s_0, s_1\}) \arrow[d, "a"] & (\{s_5, s_9\}, \emptyset) \arrow[u, "\varepsilon"] & (\{s_4, s_7\}, \emptyset) \arrow[u, "\varepsilon"] \\
& (\{s_2, s_6\}, \emptyset) \arrow[r, "\varepsilon"] & (\emptyset, \{s_2, s_6\}) \arrow[u, "d"] \arrow[ur, "c"] &
\end{tikzcd}$$

In Zustand $(\emptyset, \{s_5, s_9, s_{10}\})$ wird akzeptiert, da $s_5, s_9, s_{10} \in F$. In Zustand $(\emptyset, \{s_4, s_7, s_8\})$ wird nicht akzeptiert, da $s_8 \notin F$.

3. „AEA $\to$ AEA mit kurzen Transitionen".

   Ersetze jede Transition $\Diamond \xrightarrow{u} \bigcirc$ mit $|u| = l > 1$ durch $\Diamond \xrightarrow{u(0)} \Diamond \xrightarrow{u(1)} \cdots \xrightarrow{u(l-1)} \bigcirc$

   und jede Transition $\Box \xrightarrow{u} \bigcirc$ mit $|u| = l > 1$ durch $\Box \xrightarrow{u(0)} \Box \xrightarrow{u(1)} \cdots \xrightarrow{u(l-1)} \bigcirc$.

Zum Beweis der Korrektheit der Konstruktion 2.:

„$\Rightarrow$" Gegeben u, akzeptiert von $\mathbb{A}$. Gesucht ist ein akzeptierender Lauf des NEA.

Idee:

- Zerlege den akzeptierenden Teilbaum entsprechend der Buchstaben-Transitionen. Dann erhalten wir Zustände der Form $(\emptyset, N)$.
- Setze ε-Teilbäume in Folgen von ε-Transitionen um.

„$\Leftarrow$" Zerlege den Berechnungspfad im aufeinanderfolgende ε-Transitionen und einzelne Buchstaben-Transitionen. Rest wie oben. $\Box$

Zusatz: Es gibt eine Familie $(\mathbb{A}_n)_{n\in\mathbb{N}}$ von AEA mit $|\mathbb{A}_n| \in \mathcal{O}(n)$, so dass der minimale deterministische Automat D_n für $L(\mathbb{A}_n)$ $\Omega(2^{2^n})$ Zustände hat. Ein *UEA* ist ein AEA, der nur universelle Zustände besitzt.

$$\begin{tikzcd}
& \text{DEA} & \\
\text{NEA} \arrow[ur, "2^n"] & & \text{UEA} \arrow[ul, "2^n"'] \\
& \text{AEA} \arrow[ul, "2^n"] \arrow[ur, "2^n"'] \arrow[uu, "2^{2^n}"] &
\end{tikzcd}$$

3 Algorithmische Komplexität von automatentheoretischen Problemen

Zusammenfassung:

	DEA	NEA	UEA	AEA
Leerheitsproblem ($L(\mathbb{A}) = \emptyset$?)	NL-C (siehe (a))	NL-C (siehe (b))	PSPACE-C (siehe (f))	PSPACE-C (siehe (d))
Universalitätsproblem ($L(\mathbb{A}) = A^*$?)	NL-C (siehe (c))	PSPACE-C (siehe (e))	NL-C (siehe (f))	PSPACE-C (siehe (d))
Wortproblem ($u \in L(\mathbb{A})$?)	L	NL-C	NL-C	NP-C[1]

Das C in z. B. NL-C steht für vollständig (engl. complete).

Bemerkung 11. (a) Das Leerheitsproblem für DEA ist NL-schwierig, da Grapherreichbarkeit NL-schwierig ist (Komplexitäts-Theorie).

(b) Das Leerheitsproblem für NEA ist in NL, da Grapherreichbarkeit in NL ist.

(c) In nicht-deterministisch logarithmischen Platz lässt sich feststellen (Grapherreichbarkeit), ob es ein Wort gibt, dass *nicht* akzeptiert wird, d. h., das Universalitätsproblem ist in co-NL. Nach dem Satz von Immermann und Szelepcsényi (Vorlesung Komplexitätstheorie) gilt NL = co-NL, genauer NSPACE($S(n)$) = co-NSPACE($S(n)$) für jedes platzkonstruierbare $s(n) \geq \log n$.

(d) Leerheitsproblem und Universalitätsproblem für AEA sind polynomiell aufeinander reduzierbar, denn $L(\mathbb{A}) = \overline{L(\overline{\mathbb{A}})}$. Es reicht also zu zeigen, dass das Leerheitsproblem in PSPACE ist. Ein nicht-deterministischer Algorithmus hierfür ist der folgende:

1. Umwandlung des AEA in einen AEA mit kurzen Transitionen.
2. Simuliere die Konstruktion vom AEA mit kurzen Transitionen zum NEA:

```
(M, N) := ({s_I}, ∅)
while (M ≠ ∅ or N ⊈ F) do
    choose (non−det.) trans. ((M, N), a, (M', N'))
```

[1] oder P-C? Unklar! Siehe Übung.

$$\begin{aligned} M &:= M' \\ N &:= N' \end{aligned}$$

accept

Der Platzbedarf ist im wesentlichen zweimal die Beschreibung von (M, N), also polynomiell in der Eingabe.

Also ist das Problem in NPSPACE. Die Anwendung des Satzes von Savitch (Satz 12) liefert, dass es in PSPACE ist.

(e) Es gilt sogar, $L(\mathbb{A}_0) \cup \cdots \cup L(\mathbb{A}_{r-1}) =^? A^*$ für $\mathbb{A}_i$ DEA ist PSPACE-schwer.

Wir zeigen: $(*)$ $L(\mathbb{A}_0) \cap \cdots \cap L(\mathbb{A}_{r-1}) \neq^? \emptyset$ für $\mathbb{A}_i$ DEA ist PSPACE-schwer.

Reduktion von $L \in$ PSPACE entschieden durch DTM T auf $(*)$: Zu einem Wort u konstruiere $\mathbb{A}_0, \ldots, \mathbb{A}_{q(n)}$, so dass T das Wort u akzeptiert genau dann, wenn $L(\mathbb{A}_0) \cap \cdots \cap L(\mathbb{A}_{q(n)}) \neq \emptyset$ und zwar soll gelten $L(\mathbb{A}_0) \cap \cdots \cap L(\mathbb{A}_{q(n)}) = \{e_n\}$ mit e_n ist die Kodierung *der* akzeptierenden Berechnung von T auf u.

Kodierung einer Berechnung: Konkatenation der einzelnen Konfigurationen, normiert auf feste Länge $p(n)$.

Benutze Automaten für unterschiedliche Zwecke:

- Stelle fest, dass das Wort vom Format $u\$u\$u\ldots$ $(|u| = p(n))$ ist.
- Stelle fest, dass das Wort mit Anfangskonfiguration zu u beginnt.
- Stelle fest, dass das Wort auf akzeptierende Haltekonfiguration endet.
- Für jedes $i < p(n)$, stelle fest, dass an den Stellen $i, i + p(n), i + 2p(n), \ldots$ die richtigen Buchstaben stehen.

Dies garantiert die Korrektheit der Konstruktion. Sie ist als polynomielle Reduktion realisierbar.

(f) Gilt aufgrund der Dualität zwischen NEA und UEA (siehe Bemerkung 6).

Satz 12 (von Savitch)**.** *Für jede platzkonstruierbare Funktion $s(n) \geq \log n$ gilt NSPACE$(s(n)) \subseteq$ SPACE$(s^2(n))$.*

Beweisskizze. Divide & Conquer-Ansatz. Konstruiere eine DTM, die zu gegebener NTM „rekursiv“ bestimmt, ob man von einer Konfiguration zu einer anderen in 2^r Schritten kommt. Dann Anwendung auf die Anfangskonfiguration, Akzeptierkonfiguration und $r \in p(n)$

```
function reachable(κ, κ', r)
    for all κ'' do
        b0 := reachable(κ, κ'', r − 1)
        b1 := reachable(κ'', κ', r − 1)
        if (b0 = b1 = true) then accept
```

Zum Platzverbrauch: polynomielle Rekursionstiefe, lokale Variable mit polynomiellem Platzverbrauch, also insgesamt polynomiellen Platzverbrauch. □

4 Ausdrucksstärke / Kompaktheit

Deskriptive Komplexität von endlichen Automaten

Ziel. Wir wollen untersuchen, wie groß DEA, NEA, UEA oder AEA sein können bzw. müssen, um bestimmte Sprachen zu erkennen und diese Möglichkeiten miteinander vergleichen. Wir wollen also untersuchen, wie *kompakt* die einzelnen Automatenmodelle sind.

Wir wollen das folgende Diagramm verstehen:

$$\begin{tikzcd}
 & \mathrm{DEA} & \\
\mathrm{NEA} \arrow[ur, "\exp"] \arrow[rr, bend left, "\exp"] & & \mathrm{UEA} \arrow[ul, "\exp"'] \arrow[ll, bend left, "\exp"] \\
 & \mathrm{AEA} \arrow[uu, "2\exp"] \arrow[ul, "\exp"] \arrow[ur, "\exp"'] &
\end{tikzcd}$$

Dabei bedeutet $B \overset{\exp}{\to} A$:

- Zu jedem Automaten vom Typ B der Größe n gibt es einen äquivalenten Automaten vom Typ A mit Größe $\leq 2^{\mathrm{poly}(n)}$.
- Es gibt eine Familie $\mathbb{B}_n$ von Automaten vom Typ B mit folgenden Eigenschaften:
    1. Die Größe jedes $B \in \mathbb{B}_n$ ist polynomiell in n.
    2. Jeder Automat vom Typ A, der äquivalent zu $\mathbb{B}_n$ ist, hat mindestens 2^n Zustände.

$B \overset{2\exp}{\to} A$: analog mit doppelt exponentiell.

Beweise. 1. obere Schranken:

NEA → DEA: Potenzmengen-Konstruktion

UEA → DEA: duale Konstruktion

AEA → NEA: De-Universalisierung

AEA → UEA: Komplementierung → De-Universalisierung → Komplementierung

AEA → DEA: De-Universalisierung → Potenzmengen-Konstruktion

NEA → UEA: Potenzmengen-Konstruktion NEA → DEA, der auch ein UEA ist

UEA → NEA: De-Universalisierung UEA → DEA, der auch ein NEA ist

2. untere Schranken:

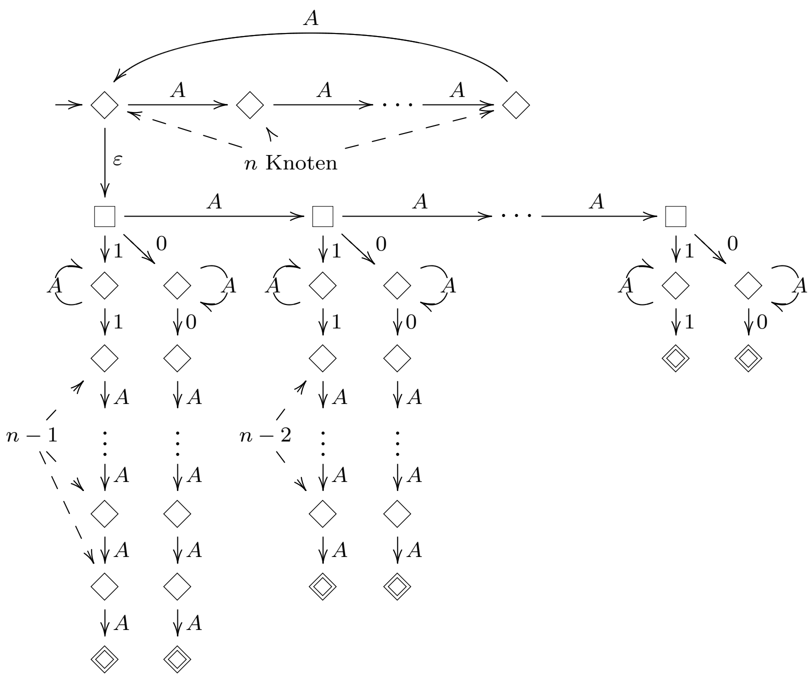
NEA → DEA: $L_n = (0+1)^*1(0+1)^n$

UEA → DEA: wegen Komplementarität

AEA → NEA: wie UEA → NEA

AEA → UEA: wie NEA → UEA

AEA → DEA: $L_n = \{u = u_0 \dots u_r \in \{0,1\}^* \mid |u_i| = n, \exists i < r : u_i = u_r\}$. Der folgende AEA erkennt L_n und besitzt $2n + 2\sum_{i=2}^{n+1} i = n^2 + 5n = \mathcal{O}(n^2)$ Zustände:



Um zu zeigen, dass ein DEA der L_n erkennt mindestens doppelt exponentiell viele Zustände besitzt benutzen wir den Satz von Nerode (Satz 13). Zu $u \in (\{0,1\}^n)^*$ sei $\alpha(u) = \{u[0,n), u[n,2n), \dots\}$. Zu jeder Menge $T \subseteq \{0,1\}^n$ wähle u_T derart, dass $\alpha(u_T) = T$.

Behauptung. $u_T \not\sim_L u_{T'}$ für $T \neq T'$.

Beweis. Sei $v \in T \triangle T'$. Dann gilt $u_T v \in L$ gdw. $u_{T'} v \notin L$. $\square$

Es gibt doppelt exponentiell viele u_T, also besitzt der minimale DEA, der L_n erkennt, mindestens doppelt exponentiell viele Zustände.

NEA → UEA: $L_n = (0+1)^*0(0+1)^n1(0+1)^* + (0+1)^*1(0+1)^n0(0+1)^*$ siehe alte Übung

UEA → NEA: wegen Komplementarität $\square$

Sei $L \subseteq A^*$. Definiere $\sim_L$ auf A^*: $u \sim_L v$ genau dann, wenn für alle $w \in A^*$ gilt, $uw \in L$ genau dann, wenn $vw \in L$.

Satz 13 (von Nerode). *1. $\sim_L$ ist eine Äquivalenzrelation,*

2. L ist regulär genau dann, wenn $\sim_L$ endlich viele Äquivalenzklassen hat und

3. falls L regulär ist, so hat der *minimale DEA für L genau so viele Zustände wie $\sim_L$ Klassen hat.*

Bemerkung 14. Es sind zahlreiche Erweiterungen der Automatenmodelle möglich:

- Erweiterungen des Wortmodells u. a. Bäume, unendliche Wörter, Kardinalzahlen, etc.
- Modellierung von Parallelität, Synchronisations-Mechanismen (erhöhen normalerweise nicht die Ausdrucksstärke, bringen aber eine größere Kompaktheit)
- In der Praxis werden oft Teilautomaten definiert und mittels Namen etc. wiederverwendet, dies erlaubt Modularität und steigert wieder die Kompaktheit.
- Zufallsgesteuerte Automaten bzw. Markow-Prozesse werden z. B. bei der Spracherkennung eingesetzt.
- Zwei-Wege-Automaten, die in beide Richtungen laufen können (erhöhen ebenfalls nicht die Ausdrucksstärke)

Teil II
Unendliche Spiele

5 Grundlegende Definitionen

Ein *Spielbrett* ist ein Tupel $\mathbb{B} = (V, V_0, V_1, E, v_I)$ wobei:

- (V, E) ist ein gerichteter Graph. Elemente von E sind die Züge.
- $V_0 \cup V_1 = V$, $V_0 \cap V_1 = \emptyset$ (V_i sind die Knoten von Spieler i, $i = 0, 1$).
- $v_I \in V$ ist der Anfangsknoten.
- Die Knoten aus V_0 bezeichnen wir mit $\Diamond$, die Knoten aus V_1 mit $\Box$.

Seien V^* und V^+ wie gehabt. V^ω ist die Menge aller unendliche Folgen über V. $V^\infty := V^+ \cup V^\omega$.
Eine *Partie* ist eine maximale Folge $\pi \in V^\infty$ mit

- $\pi(0) = v_I$ und
- für alle i mit $i + 1 < |\pi|$ gilt $(\pi(i), \pi(i+1)) \in E$.

Eine *partielle Partie* ist ein endliches Präfix einer Partie.

Beispiel 15.

$$\begin{tikzcd} {} \arrow[r] & \Diamond p_0 \arrow[r, bend left] & \boxed{p_1} \arrow[l, bend left] \arrow[r] & \Diamond p_2 \end{tikzcd}$$

Mögliche Partien sind $(p_0 p_1)^\omega$ und $(p_0 p_1)^{15} p_2$ (kurze Partie). $(p_0 p_1)^{17}$ ist keine Partie, da nicht maximal, aber eine partielle Partie.

Die *Gewinnbedingungen* für Spieler 0 sind eine Teilmenge $W \subseteq V^\infty$. Eine Partie π ist *gewinnbringend* für Spieler 0 (Spieler 0 *gewinnt* Partie π), falls $\pi \in W$. Ein *Spiel* ist das Tupel $\mathbb{G} = (\mathbb{B}, W)$. Definiere $\mathrm{occ}(u) := \{u(0), u(1), \dots\}$ und $\mathrm{inf}(u) := \{a \mid \forall i\ \exists j > i : u(j) = a\}$ (die Menge der Buchstaben, die in u unendlich oft vorkommen). Die *Erreichbarkeitsbedingungen* sind eine Teilmenge $U \subseteq V$ mit $W = V^* U V^\infty = \{u \in V^\infty \mid \mathrm{occ}(u) \cap U \neq \emptyset\}$.
Eine partielle Abbildung $\sigma\colon V^* V_0 \dashrightarrow V$ heißt *konsistent* mit E, falls $(v, \sigma(\pi v)) \in E$ für alle $\pi v \in \mathrm{dom}(\sigma)$. Eine (partielle) Partie u heißt *konform* mit $\sigma\colon V^* V_0 \dashrightarrow V$, falls für jedes i mit $i + 1 < |\pi|$ und $\pi(i) \in V_0$ gilt $\pi(i+1) = \sigma(\pi)$.
Eine *Strategie* für Spieler 0 ist eine partielle Abbildung $\sigma\colon V^* V_0 \dashrightarrow V$ mit

- für alle $u \in \mathrm{dom}(\sigma)$ gilt: $(u(|u| - 1), \sigma(u)) \in E$

- für alle $u \in V^*V_0$ gilt: wenn für alle i mit $i+1 < |u|$ und $u(i) \in V_0$ gilt: $u(i+1) = \sigma(u[0,i])$ und wenn für alle i mit $i+1 < |u|$ und $u(i) \in V_1$ gilt: $(u(i), u(i+1)) \in E$ wenn es ex. $v \in V$ mit $(u(|u|-1), v) \in E$, dann $u \in \mathrm{dom}(\sigma)$.

Fragen.

1. Hat immer einer der beiden Spieler eine Gewinnstrategie? (Nein!)
2. Wenn einer von beiden eine Gewinnstrategie hat, wie bestimmt man, wer es ist?
3. Wie sieht dann eine Gewinnstrategie aus?
4. Wie misst man die Güte einer Gewinnstrategie und wie findet man gute Gewinnstrategien?

Ein Spiel heißt *determiniert*, wenn einer der Spieler eine Gewinnstrategie hat.

6 Erreichbarkeitsbedingungen auf endlichen Graphen

Sei V endliche Menge und $W = V^*PV^\infty$ die Menge der gewinnbringenden Partien für Spieler 0, mit $P \subseteq V$.

Ziel. Fixpunktberechung zur Bestimmung der für Spieler 0 gewinnbringenden Knoten.

Dies liefert zusätzlich Determiniertheit, die Berechnung einer Gewinnstrategie und den Nachweis von Gedächtnislosigkeit (d. h. der Zug den der Spieler mit einer Gewinnstrategie in einem Knoten p macht hängt nicht von dem Weg ab, mit dem man nach p gekommen ist).

Ziel. Wir wollen

$$M_0 = \{p \mid \text{Sp. 0 hat Gew.-Str. in } G[p]\}$$

und

$$M_1 = \{p \mid \text{Sp. 1 hat Gew.-Str. in } G[p]\}$$

bestimmen und $M_0 \cup M_1 = V$ zeigen.

Definiere

$$\begin{aligned} Force_0(M) = &\{p \in V_0 \mid \exists p' \in M : (p,p') \in E\} \cup \\ &\{p \in V_1 \mid \exists p' \in M : (p,p') \in E \wedge \forall p' : (p,p') \in E \to p' \in M\}\,. \end{aligned}$$

Lemma 16. *Sei $M \subseteq M_0$. Dann gilt $Force_0(M) \subseteq M_0$.*

Beweis. Sei $p \in Force_0(M) \setminus M$.

1. Fall $p \in V_0$. Dann existiert $p' \in M$ mit $(p, p') \in E$. Sei σ Gewinnstrategie für Spieler 0 in $G[p']$. Wir konstruieren eine Gewinnstrategie σ' für Spieler 0 in $G[p]$.

$$\sigma' \colon V^*V_0 \to V$$
$$\sigma'(p) = p'$$
$$\sigma'(pp'u) = \sigma(p'u) \text{ für alle } p'u \in V^*V_0$$

Dann ist σ' eine Strategie und für jede Partie π, die mit σ' konform ist, gilt: π ist von der Form $pp'u$ mit $p'u$ konform mit σ. Also gilt $\mathrm{occ}(p'u) \cap P \neq \emptyset$, also auch $\mathrm{occ}(pp'u) \cap P \neq \emptyset$. D. h., π ist gewinnbringend für Spieler 0.

2. Fall $p \in V_1$. Dann existiert p' mit $(p, p') \in E$ und für jedes $p' \in pE$ (pE sind die Nachfolger von p bzgl. E) existiert eine Gewinnstrategie $\sigma_{p'}$ von Spieler 0 in $G[p']$. Wir definieren σ' wie folgt: Für jedes $u \in \mathrm{dom}(\sigma_{p'})$ setze $\sigma'(pu) = \sigma_{p'}(u)$, sonst σ' undefiniert. Dann ist σ' wohldefiniert und eine Strategie. Es gilt für jede mit σ' konforme Partie π:

   (a) $|\pi| \geq 2$ (Definition von $Force_0$), d. h., $\pi = pp'v$ für $p' \in pE$.

   (b) $p'v$ ist konform mit $\sigma_{p'}$, da … (gleiches Argument wie oben). □

Algorithmus zur Konstruktion von M_0.

$$\begin{aligned}
M_0^0 &= P \\
M_0^1 &= M_0^0 \cup Force_0(M_0^0) \\
M_0^2 &= M_0^1 \cup Force_0(M_0^1) \\
&\vdots \\
M_0^{j+1} &= M_0^j \cup Force_0(M_0^j)
\end{aligned}$$

irgendwann gilt wegen Monotonie und G endlich, dass $M_0^k = M_0^{k+1}$. Gebe M_0^k aus. (Algorithmus ist polynomiell und kann sogar linear implementiert werden)

Lemma 17. *Es gilt $M_0 = M_0^k$.*

Beweis. „$\supseteq$“ wegen Lemma 16.

„$\subseteq$“ Zeige dazu: Spieler 1 hat eine Gewinnstrategie für alle Positionen $p \in V \setminus M_0^k$.

Bemerkung: $P \cap (V \setminus M_0^k) = \emptyset$, da $P \subseteq M_0^0 \subseteq \cdots \subseteq M_0^k$.

Gewinnstrategie für Spieler 1 in $G[p]$ mit $p \in V \setminus M_0^k$.

$$\sigma(up') = \begin{cases} \text{undef.} & p'E = \emptyset \\ p'' \in p'E \cap (V \setminus M_0^k) & p'E \neq \emptyset \end{cases}$$

$\sigma(up')$ ist wohldefiniert, da falls $p'E \neq \emptyset$, dann auch $p'E \cap (V \setminus M_0^k) \neq \emptyset$, denn sonst $p' \in M_0^{k+1}$. Widerspruch zu $M_0^k = M_0^{k+1}$. Zu zeigen ist dann, dass σ eine Gewinnstrategie ist.

A) σ ist Strategie. Zeige: Jede partielle, mit σ konforme Partie π hat die Eigenschaft $\mathrm{occ}(\pi) \cap M_0^k = \emptyset$ (per Induktion über die Länge von π: für Knoten in V_1 siehe Definition von σ, für Knoten in V_0 benutze $M_0^k = M_0^{k+1}$).

B) σ ist Gewinnstrategie. Klar, wegen A) und $P \subseteq M_0^k$. $\square$

Damit ist der folgende Satz bewiesen.

Satz 18. *Jedes endliche Erreichbarkeitsspiel ist determiniert und der obige Algorithmus liefert M_0.*

Unterscheide im folgenden verschiedene Formen von Strategien:

allgemein Die allgemeine Form ist $\sigma\colon V^*V_0 \dashrightarrow V$.

positional Eine *positionale* (oder *gedächtnislose*) Strategie hängt nicht von vorhergehenden Spielzügen ab: $\sigma\colon V_0 \dashrightarrow V$; diese läßt sich zu einer allgemeinen Strategie ausbauen durch $\bar{\sigma}(up) = \sigma(p)$ für alle $p \in \mathrm{dom}(\sigma)$ und alle $u \in V^*$.

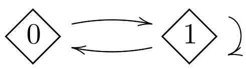
endliches Gedächtnis Eine Strategie *mit endlichem Gedächtnis* wird durch einen Halbautomaten $\mathbb{A} = (A, Q, q_I, \delta)$ modelliert, die Strategie nutzt dann die Zustandsmenge Q des Automaten: $\sigma\colon Q \times V_0 \to V$. Auch diese läßt sich zu einer allgemeinen Strategie ausbauen durch $\bar{\sigma}(up) = \sigma(\delta^*(q_I, u), p)$ für alle $u \in V^*$.

uniform Eine positionale Strategie σ ist eine *uniforme* Gewinnstrategie für Spieler 0 (oder Spieler 1), falls σ eine Gewinnstrategie für Spieler 0 (bzw. Spieler 1) ist für jede Startposition $v \in M_0$ (bzw. M_1).

uniform mit endlichem Gedächtnis Definiere *uniform* entsprechend für Strategien mit endlichem Gedächtnis.

Bemerkung 19. Eine uniforme Strategie ist auch positional, eine positionale muss aber nicht uniform sein, siehe folgendes Beispiel.

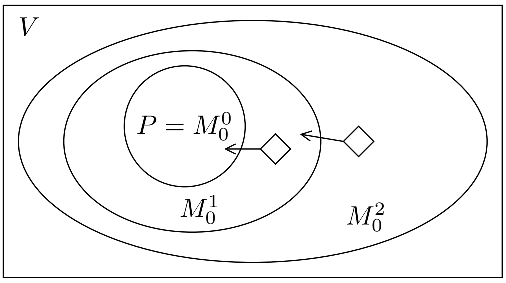
Beispiel 20. Sei folgendes Spiel gegeben: $(\{0,1\},\{0,1\},\emptyset,\{(0,1),(1,1),(1,0)\},0)$ mit einer Erreichbarkeitsbedingung $P = \{0\}$.



Der Spieler 0 gewinnt mit der positionalen Strategie $\sigma(0) = 1$, $\sigma(1) = 1$, dies ist aber keine uniforme Strategie.

Satz 21. *Es gibt für beide Spieler uniforme Gewinnstrategien.*

Beweis. Für Spieler 1 folgt dies aus obigem Beweis. Für Spieler 0:



Idee: $rang(p) = \min\{i \mid p \in M_0^i\}$. $\sigma(p) = p'$ mit $(p,p') \in E$ und $rang(p') < rang(p)$. $\square$

7 Büchi-Bedingung auf endlichen Graphen

Sei $W = (V^*P)^\omega$. Der *Attraktor* von M für Spieler 0 ist:

$Attr_0(M) =$ Ergebnis des obigen Algorithmus mit Startmenge M anstelle von P.

Beobachtung. Seien $A, B \subseteq V$ mit $A \subseteq B$. Dann gilt:

(a) $Force_0(A) \subseteq Force_0(B)$,

(b) $Attr_0(A) \subseteq Attr_0(B)$ und

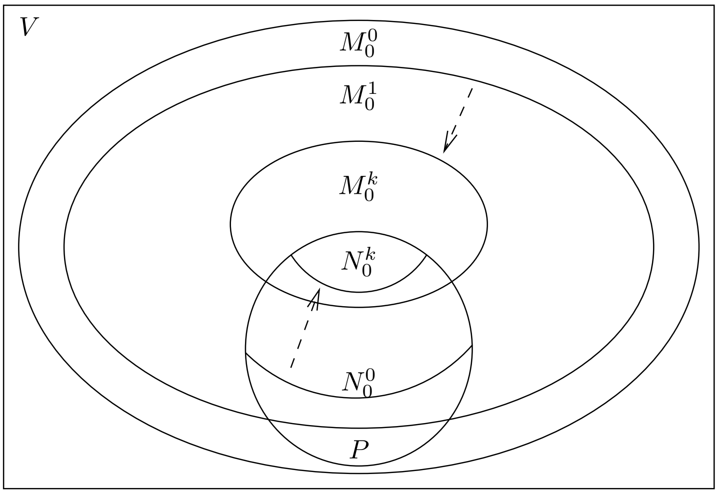
(c) $A \subseteq Attr_0(A)$.

Es gilt aber nicht zwingend $A \subseteq Force_0(A)$.

Algorithmus:

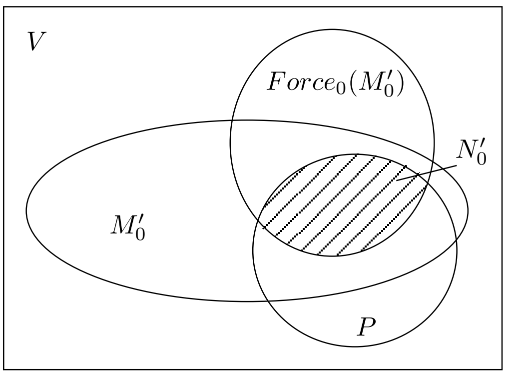
$$\begin{aligned}
M_0^0 &= Attr_0(P)\\
N_0^0 &= Force_0(M_0^0) \cap P\\
M_0^1 &= Attr_0(N_0^0)\\
N_0^1 &= Force_0(M_0^1) \cap P\\
&\vdots\\
M_0^k &= Attr_0(N_0^{k-1})\\
N_0^k &= Force_0(M_0^k) \cap P
\end{aligned}$$

Irgendwann gilt $M_0^k = M_0^{k+1}$. Gebe dann M_0^k aus. (Es gilt dann auch $N_0^k = N_0^{k+1}$)



Lemma 22. *Es gilt $M_0^k = M_0$. (Erinnerung: $M_0 = \{p \mid$ Sp. 0 hat Gew.-Str. in $G[p]\}$).*

Beweis. „$\subseteq$" Wir definieren eine positionale Strategie σ für Spieler 0. Setze $N_0' = N_0^k$ und $M_0' = M_0^k$. Es gilt dann also $N_0' = Force_0(M_0') \cap P$ und $M_0' = M_0^k = M_0^{k+1} = Attr_0(N_0^k) = Attr_0(N_0')$.



Für jedes $p \in N_0' \cap V_0$ sei $\sigma(p) =$ Position aus M_0'. Diese gibt es, da $N_0' = Force_0(M_0') \cap P$ gilt. Sei σ' eine positionale Gewinnstrategie für Spieler 0 in dem Erreichbarkeitsspiel. Dann setzen wir $\sigma(p) = \sigma'(p)$ für alle $p \in M_0' \setminus N_0'$.

„$\supseteq$" Wir zeigen, Spieler 1 hat eine uniforme Gewinnstrategie auf $V \setminus M_0'$. Konstruktion dieser Gewinnstrategie per Induktion auf den Mengen $V \setminus M_0^0$, $M_0^0 \setminus M_0^1$, ...

IA $V \setminus M_0^0$: Nehme die positionale Gewinnstrategie aus dem Erreichbarkeitsspiel (Vermeidungsspiel). P wird dann nicht besucht.

IS $M_0^i \setminus M_0^{i+1}$:

Es gilt $N_0^i = Force_0(M_0^i) \cap P$ und $M_0^{i+1} = Attr_0(N_0^i)$.

$Force_0(M_0^i) = N_0^i \cup (Force_0(M_0^i) \setminus P)$

Strategie:

- Folge der Vermeidungsstrategie für $M_0^{i+1} = Attr_0(N_0^i)$.
- Falls diese aus M_0^i herausführt, nutze Strategie nach IV, die außerhalb von M_0^i bleibt.
- Falls diese zu einer Position $p \in P$ erreicht wird, so gilt $p \notin Force_0(M_0^i)$. Ziehe im nächsten Schritt aus M_0^i heraus.

Es folgt, dass es maximal i Besuche von P gibt. □

8 Paritätsspiele auf beliebigen Graphen

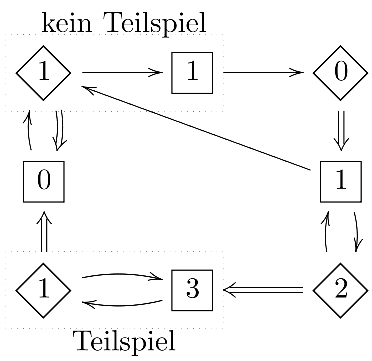
Die *Gewinnbedingung* (minimale Paritätsbedingung von Andrzej Mostowski) ist gegeben durch eine *Prioritätsfunktion* $\pi \colon V \to \mathbb{N}$ mit endlichem Wertebereich.

$$W = V^* V_1 \cup \{u \in V^\omega \mid \min\{i \mid \exists^\infty j : \pi(u(j)) = i\} \bmod 2 = 0\}$$ [2]

Ziel. Wir wollen zeigen, dass alle diese Spiele determiniert sind und beide Spieler uniforme Gewinnstrategien besitzen. Für endliche Spiele ist dieses Problem NP ∩ co-NP schwer, man kann sogar zeigen, dass es UP ∩ co-UP schwer ist.

Beispiel 23. Im folgenden Spiel gewinnt Spieler 0 von jedem Knoten aus. Eine uniforme Gewinnstrategie für Spieler 0 stellen die Doppelpfeile dar.

[2] $\exists^\infty$ heißt, es gibt unendlich viele ...



8.1 Teilspiel

Ein *Teilgraph* des Spielbrettes $B = (V, V_0, V_1, E)$ für $U \subseteq V$ ist

$$B[U] = (V \cap U, V_0 \cap U, V_1 \cap U, E \cap (U \times U)).$$

Ein *zulässiger* Teilgraph ist ein Teilgraph, falls jede *Sackgasse* (Knoten ohne ausgehenden Kanten) von $B[U]$ auch Sackgasse von B ist. Ein *Teilspiel* von B ist dann $(B[U], \pi|_U)$.

Bemerkung 24. Ein Teilspiel eines Teilspiels ist ein Teilspiel, d. h., die Teilspiel-Beziehung ist transitiv.

8.2 Fallen

Eine *σ-Falle* ($\sigma \in \{0, 1\}$) ist eine Menge $U \subseteq V$ mit

- für alle $u \in U \cap V_\sigma$ gilt $uE \subseteq U$ und
- für alle $u \in U \cap V_{1-\sigma}$ gilt $uE = \emptyset$ oder $uE \cap U \neq \emptyset$.

Lemma 25. *(a) Jede σ-Falle ist ein Teilspiel.*

(b) Für jede Familie $\{U_i\}_{i \in I}$ von σ-Fallen ist $\bigcup_{i \in I} U_i$ eine σ-Falle.

(c) Falls X eine σ-Falle ist und falls $Y \subseteq X$, so ist Y eine σ-Falle in G genau dann, wenn Y eine σ-Falle in $G[X]$ ist.

Beweis. (a) Sei U eine σ-Falle. Sei $u \in U$. Falls $u \in U \cap V_\sigma$. Dann $uE \subseteq U$. Also: Wenn kein Nachfolger in U, so auch kein Nachfolger in V. Falls $u \in U \cap V_{1-\sigma}$. Dann: Falls $uE \neq \emptyset$, so $uE \cap U \neq \emptyset$, also keine Sackgasse in $B[U]$, falls keine Sackgasse in B.

(b) Trivial.

(c) „$\Leftarrow$“ Angenommen Y ist σ-Falle in $G[X]$. Sei $y \in Y \cap V_\sigma$. Dann, da Y σ-Falle in $G[X]$ gilt $y(E \cap (X \times X)) \subseteq Y \cap X = Y$. Für u mit $(y,u) \in E$, $u \notin X$ gilt: Widerspruch, da $y \in X \cap V_\sigma$ und X σ-Falle. Sei $y \in Y \cap V_{1-\sigma}$. Da Y σ-Falle in $G[X]$ gilt: $\underbrace{y(E \cap (X \times X))}_{\subseteq yE} \cap Y \neq \emptyset$.

„$\Rightarrow$“ Angenommen Y ist eine σ-Falle in G. Sei $y \in Y \cap (V_\sigma \cap X) = Y \cap V_\sigma$. Dann gilt, da Y σ-Falle in G: $yE \subseteq Y$, also auch $y(E \cap (X \times X)) \subseteq Y \cap X = Y$. Sei $y \in Y \cap (V_{1-\sigma} \cap X) = Y \cap V_{1-\sigma}$. Dann existiert $y' \in Y$ mit $(y,y') \in E$. Dann gilt aber auch $(y,y') \in E \cap (X \times X)$. □

Lemma 26. *Sei $U \subseteq V$. Dann gilt:*

a) $V \setminus Attr_\sigma(U)$ ist eine σ-Falle.

b) Falls U eine σ-Falle ist, so ist $Attr_{1-\sigma}(U)$ eine σ-Falle.

c) U ist eine σ-Falle in G genau dann, wenn $Attr_\sigma(V \setminus U) = V \setminus U$.

d) $Attr_\sigma(U) = V \setminus X$, wobei X die größte σ-Falle in $V \setminus U$ ist.

Beweis. a) Trivial.

b) Trivial.

c) Trivial.

d) Sieht man leicht ..., ist aber nicht trivial. □

8.3 Paradies

Eine Menge $U \subseteq V$ ist ein σ-*Paradies*, falls U eine $(1-\sigma)$-Falle ist und es eine uniforme positionale Strategie f_σ für σ auf U gibt, die auf U gewinnbringend ist.

Bemerkung 27. Dann gilt $U \subseteq M_\sigma$.

Lemma 28. *a) Ist U ein σ-Paradies, so auch $Attr_\sigma(U)$.*

b) Ist $\{U_i\}_{i \in I}$ eine Familie von σ-Paradiesen, so auch $\bigcup_{i \in I} U_i$.

Beweis. a) Da U ein σ-Paradies ist, ist U insbesondere eine $1-\sigma$-Falle. $Attr_\sigma(U)$ ist eine $(1-\sigma)$-Falle, nach Lemma 26 b). Gewinnstrategie

$$f'_\sigma(u) = \begin{cases} g(u) & \text{für } u \in Attr_\sigma(u) \setminus U,\ g \text{ Gew.-Str. aus dem Attr.-Spiel} \\ f_\sigma(u) & \text{sonst, mit } f_\sigma \text{ Gew.-Str aus dem Paradies} \end{cases}$$

b) $\bigcup U_i$ ist eine $(1-\sigma)$-Falle nach Lemma 26 b). Nach dem Wohlordnungsprinzip gibt es eine Wohlordnung $<$ auf I. Nach den Vorraussetzung gibt es für jedes U_i eine uniforme Gewinnstrategie f_i auf U_i. Definiere nun eine uniforme Strategie für $\bigcup U_i$ wie folgt:

$$f(u) = f_j(u), \text{ für } j \text{ minimal mit } u \in U_j.$$

Beweis für die Korrektheit: In jeder konformen Partie gibt es einen Punkt ab dem alle Knoten in U_i liegen und nach f_i gespielt wird. □

8.4 Exkurs: Auswahlaxiom

Auswahlaxiom. Zu jeder Menge von Mengen $\{M_i\}_{i\in I}$ gibt es eine Funktion $f\colon I \to \bigcup_{i\in I} M_i$, so dass für alle $i \in I$ gilt: $f(i) \in M_i$.

$(M, <)$ ist eine *Wohlordnung*, wenn $(M, <)$ eine lineare Ordnung (oder Totalordnung, Kette) ist und jede nichtleere Teilmenge von M ein kleinstes Element besitzt. D. h. es gibt keine unendliche Folge $m_1 > m_2 > \ldots$.

Wohlordnungsprinzip. Zu jeder Menge M gibt es eine Relation $<$, so dass $(M, <)$ eine Wohlordnung ist.

Das Wohlordnungsprinzip folgt aus dem Auswahlaxiom (und umgekehrt).

Literatur: z. B. „Naive Mengenlehre“ von Paul R. Halmos

8.5 Paritätsspiele sind determiniert

Satz 29. *Ist G ein Paritätsspiel, so gibt es ein 0-Paradies P_0 und ein 1-Paradies P_1 mit $P_0 \cup P_1 = V$.*

Bemerkung 30. Für den Beweis betrachten wir Max-Paritätsspiele, bei denen die höchste Priorität ausschlaggebend ist.

Lemma 31. *Wenn* $\mathrm{Bild}(\pi) = \{0\}$*, so gibt es solche Paradiese.*

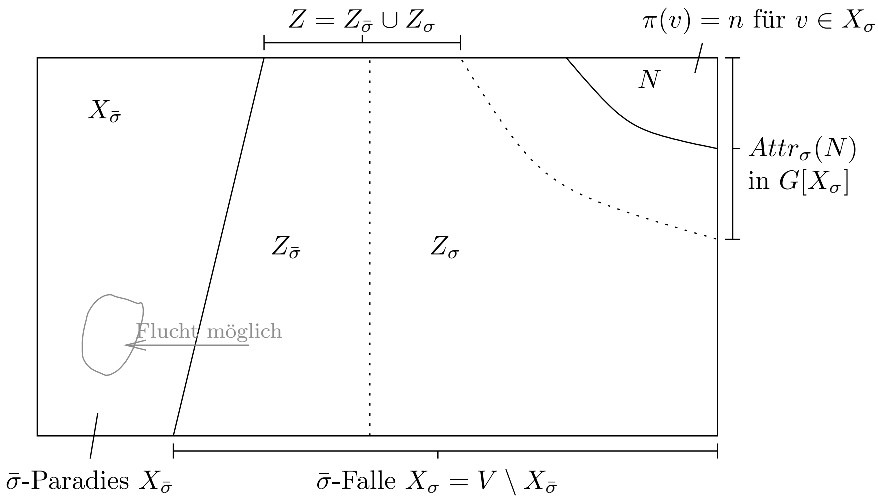
Beweis. Sei D die Menge der Sackgassen für Spieler 0 und $A = \mathit{Attr}_1(D)$. Wir zeigen, dass $V \setminus A$ und A die beiden gesuchten Paradiese sind.

1. Nach Lemma 26 a) ist $V \setminus A$ eine 1-Falle. Auf dieser hat Spieler 0 nach den Überlegungen zu Erreichbarkeitsspielen eine positionale Strategie, die alle Knoten aus D vermeidet, d.h. er gewinnt dort.

2. Spieler 1 hat nach den Überlegungen zu Erreichbarkeitsspielen eine uniforme Gewinnstrategie auf A. Da D aus Sackgassen besteht, ist A auch 0-Falle. □

Bezeichne mit $Attr_\sigma^G(U)$ den σ-Attraktor von U im Spiel G.

Lemma 32. *Sei $n = \max(\mathrm{Bild}(\pi)) \geq 1$. Sei $\sigma = n \bmod 2$, d. h., σ ist der Spieler, der mit n gewinnt. Sei $\bar{\sigma} = 1 - \sigma$ und $X_{\bar{\sigma}} \subseteq V$ ein $\bar{\sigma}$-Paradies in G, so dass $X_\sigma = V \setminus X_{\bar{\sigma}}$ eine $\bar{\sigma}$-Falle in G ist. Sei $N = \{v \in X_\sigma \mid \pi(v) = n\}$ und $Z = X_\sigma \setminus Attr_\sigma^{G[X_\sigma]}(N)$.*



Falls $Z = Z_\sigma \uplus Z_{\bar{\sigma}}$ ist mit σ- bzw. $\bar{\sigma}$-Paradiesen Z_σ und $Z_{\bar{\sigma}}$ bzgl. $G[Z]$. Dann gilt folgendes:

(a) $X_{\bar{\sigma}} \cup Z_{\bar{\sigma}}$ ist ein $\bar{\sigma}$-Paradies in G.

(b) Falls $Z_{\bar{\sigma}} = \emptyset$, dann ist X_σ ein σ-Paradies in G.

Beweis. $G[X_\sigma]$ ist eine $\bar{\sigma}$-Falle, nach Lemma 26 (a), also ein Teilspiel von G. Z ist das Komplement von $Attr_\sigma^{G[X_\sigma]}$ in X_σ, nach Lemma 25 (a), also eine σ-Falle in $G[X_\sigma]$. Nach Lemma 26 (a) ist $G[Z]$ also ein Teilspiel von $G[X_\sigma]$ und wegen der Transitivität der Teilspielrelation (Bemerkung **??**) auch ein Teilspiel von G.

(a) 1. Nach der Definition eines Paradieses wissen wir, dass $Z_{\bar{\sigma}}$ eine σ-Falle in $G[Z]$ ist. Z ist σ-Falle in $G[X_\sigma]$, $Z_{\bar{\sigma}} \subseteq Z$ und $Z_{\bar{\sigma}}$ ist eine σ-Falle in $G[Z]$. Mit Lemma 26 (c) folgt also, dass $Z_{\bar{\sigma}}$ auch eine σ-Falle in $G[X_\sigma]$ ist.

   Nach Lemma 26 (b) ist $X_{\bar{\sigma}} \cup Z_{\bar{\sigma}}$ eine σ-Falle in G.

2. Wir betrachten eine Strategie, die sich aus Gewinnstrategien für die beiden $\bar{\sigma}$-Paradiese $X_{\bar{\sigma}}$ und $Z_{\bar{\sigma}}$ zusammensetzt. Sei u eine Partie konform mit dieser Strategie. Dann gibt es zwei Möglichkeiten:

i. Sei $u(i) \in Z_{\bar\sigma}$ für alle i. Dann ist u eine Partie konform mit der Gewinnstrategie für $Z_{\bar\sigma}$ in $G[Z]$.
ii. Sei $u(i) \in X_{\bar\sigma}$ für ein i. Sei i_0 das kleinste solche. Dann ist $u[i_0, \infty)$ eine Partie konform mit der Gewinnstrategie für $X_{\bar\sigma}$.

In beiden Fällen gewinnt Spieler $\bar\sigma$ auf $X_{\bar\sigma} \cup Z_{\bar\sigma}$.

(b) 1. Nach Voraussetzung ist X_σ eine $\bar\sigma$-Falle in G.

2. Gewinnstrategie für Spieler σ in einem Knoten $v \in X_\sigma \cap V_\sigma$:

i. Falls $v \in N$ ist, so zieht Spieler σ zu irgend einem Nachfolger in X_σ – dieser existiert, da X_σ eine $\bar\sigma$-Falle ist.
ii. Falls $v \in Attr_\sigma^{G[X_\sigma]}(N) \setminus N$ ist, so zieht Spieler σ gemäß der Attraktor-Strategie (d. h. in Richtung N).
iii. Falls $v \in Z_\sigma$ ist, so zieht Spieler σ gemäß der Paradies-Strategie für Z_σ.

Es ist leicht zu zeigen, dass dies eine Gewinnstrategie auf X_σ für Spieler σ ist. □

Beweis des Satzes 29. Beweis per Induktion über $n = \max(\mathrm{Bild}(\pi))$.

IA Falls $n = 0$, so benutze Lemma 31.

IS Sei $n \geq 1$, sei $\sigma = n \bmod 2$. Sei $\{W_i\}_{i \in I}$ die Menge aller $\bar\sigma$-Paradiese in G. Diese Familie ist nicht leer, da $\emptyset$ dazugehört. Nach Lemma 28 (b) ist $X_{\bar\sigma} = \cup_{i \in I} W_i$ ein $\bar\sigma$-Paradies, also das maximale. Sei $X_\sigma = V \setminus X_{\bar\sigma}$. Nach Lemma 28 (a) ist $Attr_{\bar\sigma}^G(X_{\bar\sigma})$ ein $\bar\sigma$-Paradies in G. Da $X_{\bar\sigma}$ das größte ist, gilt $X_{\bar\sigma} = Attr_{\bar\sigma}^G(X_{\bar\sigma})$. Also ist X_σ das Komplement eines $\bar\sigma$-Attraktors und somit (nach Lemma 25 (a)) eine $\bar\sigma$-Falle in G.

Wir betrachten nun – wie in Lemma 32 – $N = \{v \in X_\sigma \mid \pi(v) = n\}$ und $Z = X_\sigma \setminus Attr_\sigma^{G[X_\sigma]}(N)$.

Falls $Z = \emptyset$, so lässt sich $G[Z]$ trivialerweise in ein σ- und ein $\bar\sigma$-Paradies bzgl. Z partitionieren, nämlich $Z_{\bar\sigma} = Z_\sigma = \emptyset$. Ansonsten gilt, wie im Beweis von Lemma 32 gesehen, dass $G[Z]$ ein Teilspiel von G ist. Es gilt $\max(\mathrm{Bild}(\pi|_Z)) < n$ und nach Induktionsvoraussetzung lässt sich $G[Z]$ also in ein σ- und ein $\bar\sigma$-Paradies bzgl. Z partitionieren. Seien diese Z_σ und $Z_{\bar\sigma}$.

Angenommen $Z_{\bar\sigma} \neq \emptyset$, so wäre nach Lemma 32 (a) $X_{\bar\sigma} \cup Z_{\bar\sigma}$ ein $\bar\sigma$-Paradies in G, größer als $X_{\bar\sigma}$. Widerspruch zur Maximalität von $X_{\bar\sigma}$. Also gilt $Z_{\bar\sigma} = \emptyset$.

Nach Lemma 32 (b) ist dann X_σ ein σ-Paradies in G. □

Fazit. Der Beweis des Satzes 29, dass jedes Paritätsspiel determiniert ist und uniforme Strategien zulässt, liefert auch ein rekursives Verfahren zum Lösen von Paritätsspielen.

8.6 Komplexität

ParityGame: Gegeben ein endliches Paritätsspiel (jetzt wieder ein Min-Paritätsspiel), hat Spieler 0 eine Gewinnstrategie?

Satz 33. ParityGame $\in$ *NP* $\cap$ *co-NP.*

Beweis. NP-Algorithmus: Rate positionale Strategie für Spieler 0 und überprüfe, ob dies eine Gewinnstrategie ist. Noch zu zeigen ist, dass die Überprüfung in Polynomzeit möglich ist.
co-NP-Algorithmus: Teste, wie oben für Spieler 0, ob Spieler 1 eine Gewinnstrategie hat. □

WinningStrategy: Gegeben ein Spiel $G = (V, V_0, V_1, E, v_0, \pi)$ und $\sigma\colon V_0 \to V$ positionale Strategie. Frage: Ist σ eine Gewinnstrategie für Spieler 0?

Idee. für einen Polynomzeitalgorithmus, der WinningStrategy löst. Betrachte $H = (V, F)$ mit $F = (E \cap (V_1 \times V)) \cup \sigma$ und für jedes ungerade $i \in \text{Bild}(\pi)$ auch

$$H_i = (\underbrace{V \setminus \{v \in V \mid \pi(v) < i\}}_{=:W_i}, F \cap (W_i \times W_i)).$$

Dann gilt: σ ist keine Gewinnstrategie für Spieler 0 genau dann, wenn es gibt $i \in \text{Bild}(\pi)$ und starke Zusammenhangskomponente C in H_i mit folgenden Eigenschaften:

- C ist von v_0 in H erreichbar
- C enthält v mit $\pi(v) = i$

Laufzeit: $|V| \cdot (|E| + |V|)$.

Bemerkung 34.
1. Es gibt Algorithmen, die sehr effizient sind, von denen man aber weder zeigen kann, dass ihre Laufzeit polynomiell ist, noch, dass sie superpolynomiell ist. Stichwort: policy iteration.
2. Der beste bekannte Algorithmus, dessen Laufzeit abgeschätzt werden kann, hat Laufzeit $\mathcal{O}(d \cdot m \cdot (\frac{n}{\lfloor d/2 \rfloor})^{\lfloor d/2 \rfloor})$ wobei d die Anzahl der Prioritäten, m Kanten, n Knoten und braucht polynomiellen Platz (M. Jurdzinski). Alternative von H. Seidl mit höherer Platzkomplexität.

3. Es gibt Algorithmen, die auf bestimmten Graphklassen, in polynomieller Zeit laufen (beschränkte Baumweite, beschränkte Hyper-Baumweite, entanglement, ...).

4. Beste Kompl.-theor. Schranke: UP $\cap$ co-UP (UP – eindeutig, max. ein akz. Lauf) (M. Jurdzinski)

8.7 Mean Payoff Games

Ein *mean payoff game* ist ein Tupel $(V_0, V_1, E, v_0, \nu, d, w)$ mit

- V_0, V_1, E enthält keine Sackgassen.
- $\nu \in \mathbb{N}$ ist ein Schwellwert.
- $d \in \mathbb{N}$ ist das maximale Gewicht.
- $w \colon E \to \{-d, \dots, d\}$.

Spieler 0 gewinnt Partie π genau dann, wenn

$$\liminf_{t\to\infty} \underbrace{\frac{1}{t} \sum_{i=0}^{t-1} w(\pi(i), \pi(i+1))}_{\in[-d,d]} \geq \nu.$$

Eine Variante, genannt *discounted*, führt ein $\lambda \in (0,1)$ ein. Dann gilt folgende Gewinnbedingung:
Spieler 0 gewinnt Partie π genau dann, wenn

$$\underbrace{(1-\lambda) \sum_{i=0}^{\infty} \lambda^i w(\pi(i), \pi(i+1))}_{\in[-d,d]} \geq \nu.$$

(Die Summe konvergiert, da $\sum \lambda^i$ absolut gegen $\frac{1}{1-\lambda}$ konvergiert)

Es ist leicht, Paritätsspiele in (discounted) mean payoff games zu übersetzen.
Also: Algorithmus mit erwarteter Laufzeit $2^{\mathcal{O}(\sqrt{m})}$.
Die Komplexität dieser Spiele ist noch offen.

8.8 Neue Gewinnbedingungen

Voraussetzung: alle Knoten sind gefärbt, d. h., es gibt eine *Färbung* $c\colon V \to C$ mit $C := \{0, \dots, d-1\}$. c kann in natürlicher Weise auf V^ω erweitert werden, durch $c(\pi) = \{c(v) \mid \exists i \in \mathbb{N} : v = \pi(i)\}$ für $\pi \in V^\omega$.

Rabin-Bedingung: $\{(L_0, U_0), \dots, (L_{r-1}, U_{r-1})\}$ mit $L_i, U_i \subseteq C$. Die Partie π ist gewinnbringend für Spieler 0, falls $i < r$ existiert, so dass $L_i \cap \inf(c(\pi)) = \emptyset$ und $U_i \cap \inf(c(\pi)) \neq \emptyset$.

Streett-Bedingung: $\{(L_0, U_0), \dots, (L_{r-1}, U_{r-1})\}$ mit $L_i, U_i \subseteq C$. Die Partie π ist gewinnbringend für Spieler 0, falls für alle $i < r$ gilt: falls $L_i \cap \inf(c(\pi)) \neq \emptyset$, so $U_i \cap \inf(c(\pi)) \neq \emptyset$.

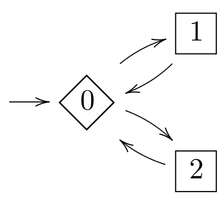
Muller-Bedingung: $\mathcal{F} \subseteq 2^C$. Die Partie π ist gewinnbringend für Spieler 0, falls $\inf(c(\pi)) \in \mathcal{F}$.

leads-to-Bedingung: $\{(L_0, U_0), \dots, (L_{r-1}, U_{r-1})\}$ mit $L_i, U_i \subseteq C$. Die Partie π ist gewinnbringend für Spieler 0, falls für alle $i < r$ und $j < \infty$ gilt, wenn $c(\pi(j)) \in L_i$, so existiert $j' > j$ mit $c(\pi(j')) \in U_i$.

Gegeben sei G mit minimaler Paritätsbedingung $\pi\colon V \to \{0, \dots, m-1\}$.
als Rabin-Bedingung: $\{(\emptyset, \pi^{-1}(0)), (\pi^{-1}(1), \pi^{-1}(2)), (\pi^{-1}(\{1,3\}), \pi^{-1}(4)), \dots\}$
oder $\{(\emptyset, \pi^{-1}(0)), (\pi^{-1}(\{0,1\}), \pi^{-1}(\{0,1,2\})), (\pi^{-1}(\{0,1,2,3\}), \pi^{-1}(\{0,1,2,3,4\})), \dots\}$
(Rabin chain condition) … Umwandlung in Farben durch weglassen der π^{-1}.

Dualität zwischen Rabin- und Streett-Bed.: π erfüllt Rabin-Bed. $\{(L_0, U_0), \dots, (L_{r-1}, U_{r-1})\}$ nicht gdw. π erfüllt die folgende Streett-Bed. $\{(U_0, L_0), \dots, (U_{r-1}, L_{r-1})\}$.

Beobachtung. Es gibt Streett-Spiele, die Spieler 0 gewinnen kann, aber nicht mit einer positionalen Strategie. Zum Beispiel folgendes Spiel



mit Streett-Gewinnbedingung $\{(V, \{1\}), (V, \{2\})\}$.
Analoge Aussage für Rabin-Spiele.

Ziel. Reduziere Rabin-, Streett- und Muller-Spiele auf Paritätsspiele.

Rabin $\to$ Muller: $\mathcal{F} = \{F \mid \exists i : F \cap L_i = \emptyset \wedge U_i \cap F \neq \emptyset\}$.

Betrachte die folgende Datenstruktur *latest appearance record (with hit):*
Sei C ein Alphabet und $¢$ ein neues Symbol. Sei I_C die Menge aller Wörter über $C \cup \{¢\}$, in denen mindestens $¢$ vorkommt und jeder Buchstabe höchstens ein mal. Definiere dann update: $I_C \times C \to I_C$ durch

$$\text{update}(u¢v, c) \mapsto \begin{cases} w_0¢wc & uv = w_0cw, c \in \alpha(uv) \\ ¢uvc & \text{sonst} \end{cases}$$

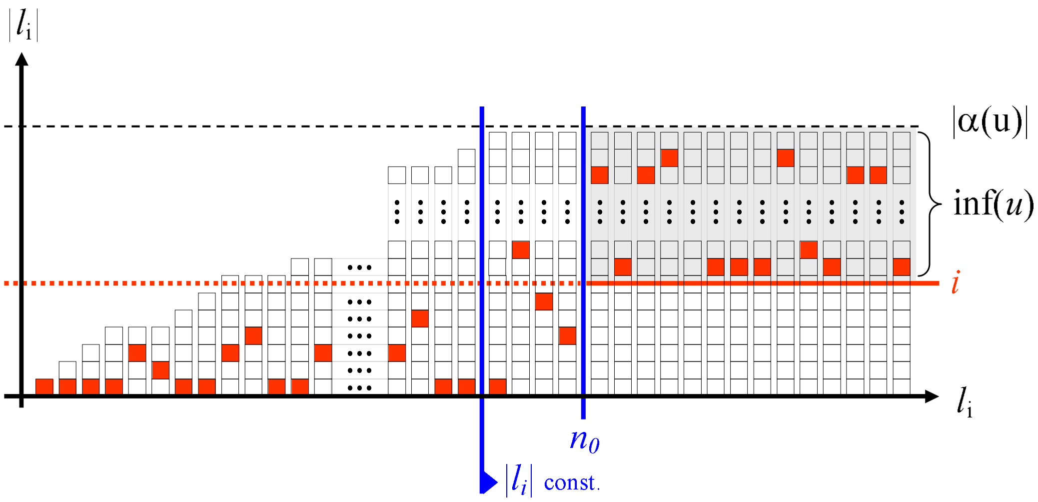
Dabei sei $\alpha(x)$ die Menge der in x vorkommenden Buchstaben.

Lemma 35. *Sei $u \in C^\omega$ und $l_0, l_1, \ldots$ definiert durch $l_0 = ¢$ und $l_{i+1} = \text{update}(l_i, u(i))$. Sei p_i diejenige Position mit $l_i(p_i) = ¢$. Dann gilt für jedes j:*

(a) $\alpha(l_{j+1}) \setminus \{¢\} = \alpha(u[0, j])$.

(b) Für $c \neq c' \in C$ gilt: Falls $l_{j+1} = w_0c'w_1cw_2$ ist, so existieren v_0, v_1, so dass $u[0, j] = v_0cv_1$ ist mit $c' \notin \alpha(v_1)$.

Außerdem: Sei $i = \liminf \{p_j \mid j \geq 0\}$, d.h. der kleinste Index, an dem der Marker $¢$ unendlich oft auftritt. Sei zudem n_0 die kleinste Position mit $|l_{n_0}| = |l_{n_0+1}| = |l_{n_0+2}| \ldots$ sowie $\min\{p_j \mid j > n_0\} \geq i$.



(c) Für jedes $n \geq n_0$ mit $p_n = i$ gilt dann: $\alpha(l_n[i + 1, |l_n|)) = \text{inf}(u)$.

(d) Sei $n'_0 \geq n_0$ minimal mit $p_{n'_0} = i$. Dann gilt für alle $n \geq n'_0$ mit $|\text{inf}(u)| = k$, dass $\alpha(\text{suffix}_{k+1}(l_n)) = \text{inf}(u) \cup \{¢\}$ ist.

Beweis. (a) per Induktion über j

(b) per Induktion über j:

IA trivial

IS Sei $c_0 = u(j+1)$. Da l_{j+1} und l_{j+2} ohne $¢$ und c_0 gleich sind, brauchen wir nur die Fälle $c = c_0$ und $c' = c_0$ zu betrachten.

1. Der Fall $c = c_0$. Sei $l_{j+2} = w_0 c' w_1 c_0 w_2$, dann folgt $w_2 = \varepsilon$. Wähle $v_0 = u[0, j]$ und $v_1 = \varepsilon$.
2. Der Fall $c' = c_0$. Sei $l_{j+1} = w_0 c' w_1 c_0 w_2$, das ergibt aber einen Widerspruch, da l_{j+2} auf c_0 enden muss.

(c) Sei n_1 minimale Position mit $\alpha(u[n_1, \infty)) = \inf(u)$, und sei n_2 minimal mit $\alpha(u[n_1, n_2)) = \inf(u)$.

Für alle $j \geq n_2$ gilt dann: $\text{suffix}_{k+1}(l_j) = \inf(u) \cup \{¢\}$, dies folgt aus b).

(d) ... □

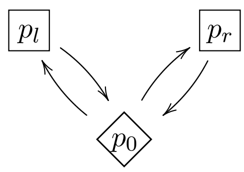
Lemma 36. *Sei u wie oben und $l = l_0 l_1 \ldots$. Sei $\mathcal{F}$ eine Muller-Bedingung über C. Dann sind äquivalent:*

(i) u ist gewinnbringend für Spieler 0 bzgl. $\mathcal{F}$.

(ii) l ist gewinnbringend bezüglich der folgenden Rabin-Bedingung Ω für Spieler 0:

$$\Omega = \{\,(L_F, U_F) \mid F \in \mathcal{F}\}$$
$$L_F = \{\,w_0 ¢ w_1 \in I_C \mid |w_1| > |F|\}$$
$$U_F = \{\,w_0 ¢ w_1 \in I_C \mid \alpha(w_1) = F\}$$

Beispiel 37. Sei $\mathcal{F} = \{p_0, p_l, p_r\}$ für folgendes Spiel



Spieler 0 hat eine Gewinnstrategie λ mit Gedächtnis $A = (\{p_0, p_l, p_r\}, \{l, r\}, l, \delta)$, wobei δ gegeben ist durch folgende Transitionstabelle:

δ	p_0	p_l	p_r
l	l	l	r
r	r	r	l

λ ist dann gegeben durch $\lambda(l, p_0) = p_r$ und $\lambda(r, p_0) = p_l$.

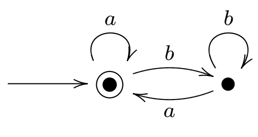
Satz 38. *In Rabin-Spielen (bzw. Streett-Spielen) hat Spieler 0 (bzw. Spieler 1) eine uniforme Strategie.*

Teil III
Endliche Automaten auf unendlichen Wörtern

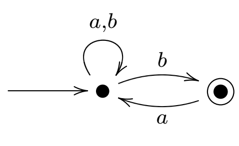
9 Einführung

Ein *endlicher Automat auf unendlichen Wörtern*, auch *endlicher ω-Automat* gennannt, ist ein Tupel $\mathbb{A} = (A, Q, q_I, \Delta, \Omega)$ mit der *Akzeptierbedingung* Ω, welche eine Büchi-, Rabin-, Streett- oder Muller-Bedingung ist. $\Delta \subseteq Q \times A \times Q$.
Büchi-Automaten sind solche mit $\Omega = F \subseteq Q$. Ein Lauf eines Büchi-Automaten *erfüllt* F, falls ein Zustand aus F unendlich oft besucht wird.
$\mathbb{A}$ *akzeptiert* $u \in A^\omega$, falls es einen Lauf von $\mathbb{A}$ auf u gibt, der Ω erfüllt. $L(\mathbb{A}) = \{u \in A^\omega \mid \mathbb{A} \text{ akz. } u\}$.

Beispiel 39. (a) Die Sprache $L = \{u \in \{a, b\}^\omega \mid a \in \inf(u)\}$ wird von folgendem *det.* Büchi-Automaten erkannt:

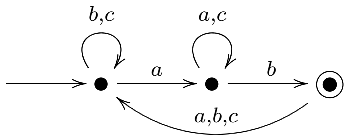


(b) Die Sprache $L = \{u \in \{a, b\}^\omega \mid \{a, b\} = \inf(u)\}$ wird von folgendem *nicht-det.* Büchi-Automaten erkannt:

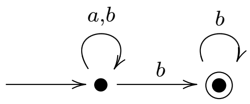


Es gibt aber auch det. Büchi-Automaten, die L erkennen.

(c) Die Sprache $L = \{u \in \{a,b,c\}^\omega \mid a, b \in \inf(u)\}$ wird von folgendem *det.* Büchi-Automaten erkannt:



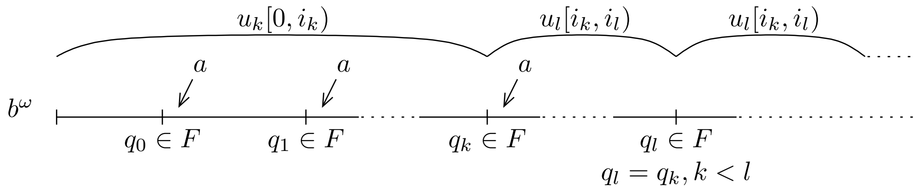
(d) Die Sprache $L = \{u \in \{a,b\}^\omega \mid a \text{ endlich oft}\}$ wird von folgendem *nicht-det.* Büchi-Automaten erkannt:



Es gibt *keinen* det. Büchi-Atuomaten, der L erkennt!

Satz 40. *Die Sprache* $L = \{u \in \{a,b\}^\omega \mid a \notin \inf(u)\}$ *wird von* keinem *deterministischen Büchi-Automaten erkannt, aber von deterministischen Rabin-, Muller- oder Streett-Automaten.*

Beweis. Durch ein Cut-and-Paste-Argument, oder auch Pumping-Argument.

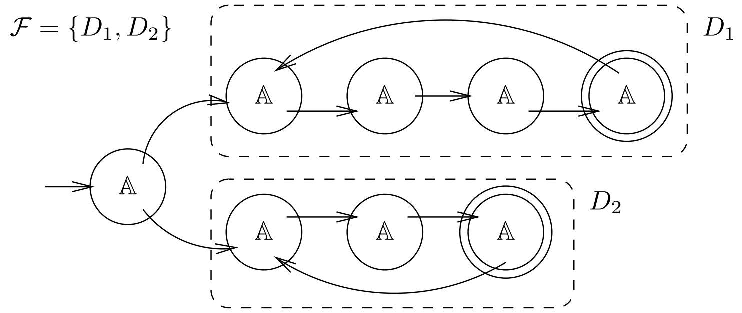


Angenommen es gibt einen det. Büchi-Automaten $\mathbb{A}$, der L erkennt. $\mathbb{A}$ akzeptiert dann insbesondere $u_0 = b^\omega$. In der Berechnung von $\mathbb{A}$ auf u_0 wird irgendwann, etwa nach i_0 Schritten, ein Zustand $q_0 \in F$ erreicht. Setze $u_1 = u_0[0, i_0)ab^\omega = b^{i_0}ab^\omega$. u_1 ist auch in L, also wird $\mathbb{A}$ auch bei der Berechnung auf u_1 nach etwa $i_1 > i_0$ Schritten in einem Endzustand $q_1 \in F$ landen. Setze wiederum $u_2 = u_1[0, i_1)ab^\omega$. Dieser Prozess lässt sich wiederholen, bis einer der Endzustände $q_0, q_1, \ldots$ doppelt auftritt, d. h. $q_k = q_l$ für $k < l$. Dann folgt, dass $\mathbb{A}$ das Wort $u_k[0, i_k)(u_l[i_k, i_l))^\omega$, welches unendlich viele a enthält, erkennt. Ein Widerspruch. $\square$

Definition 41. $L \subseteq A^\omega$ heißt *regulär*, falls L von einem endlichen ω-Automaten erkannt wird.

Satz 42. *Falls L regulär ist, so wird L von einem nicht-deterministischen Büchi-Automaten erkannt.*

Beweis. Es reicht zu zeigen: L wird von einem ndet. Muller-Automaten erkannt $\Rightarrow$ L wird von einem ndet. Büchi-Automaten erkannt.



Sei $\mathbb{A} = (A, Q, q_I, \Delta, \mathcal{F})$ ein Muller-Automat. Daraus kann man folgenden Büchi-Automaten konstruieren: $\mathbb{A}' = (A, Q', q_I', \Delta', F')$ mit

$$Q' = Q \cup (Q \times \mathcal{F} \times \{0, \dots, |Q|\})$$

$$q_I' = q_I$$

$$\Delta' = \Delta \cup \{(q, a, (q', F, 0)) \mid (q, a, q') \in \Delta, F \in \mathcal{F}\} \cup \bigcup_{F \in \mathcal{F}} \Delta_F$$

Sei $F = \{q_0, \dots, q_{r-1}\} \in \mathcal{F}$. Definiere

$$\begin{aligned}\Delta_F = &\{((q, F, i), a, (q', F, i)) \mid (q, a, q') \in \Delta, q, q' \in F, q' \neq q_i, i < r\} \\ &\cup \{((q, F, i), a, (q', F, i + 1)) \mid (q, a, q') \in \Delta, q, q' \in F, q' = q_i, i < r\} \\ &\cup \{((q, F, r), a, (q', F, 0)) \mid (q, a, q') \in \Delta, q, q' \in F\}\end{aligned}$$

$$F' = \{(q, F, |F|) \mid F \in \mathcal{F}\}$$

$\square$

10 Konstruktion von Muller-Schupp

Ziel. Zu gegebenem (nicht-det.) Büchi-Automaten $\mathcal{A} = (A, Q, q_I, \Delta, F)$ wollen wir einen äquivalenten deterministischen Rabin-Automaten $\mathcal{D} = (A, Q', q_I', \delta, \Omega)$ konstruieren.

Idee. Sei S eine endliche Menge von Bäumen, die den bisherigen Berechnungsbaum kodieren.
Ein Zustand s ist gegeben durch:

- Endlicher Binärbaum (keinen Nachfolger oder zwei Nachfolger, mit geordneten Nachfolgern)
- Knoten aus einer Menge N (Namen) mit Anzahl $|N| = 3\,|Q|$.
- Beschriftung aller Knoten durch Farben grün, gelb und rot.
- Beschriftung der Blätter durch Mengen von Zuständen, alle p. w. disjunkt, d. h., Anzahl der Blätter $\leq |Q|$, also Anzahl der Knoten $\leq 2\,|Q| - 1$, da Anzahl der inneren Knoten $\leq |Q| - 1$.

Anfangszustand: $\{q_I\}, \text{rot}$

Übergangsfunktion: Wird beschrieben durch Alg., der aus gegebenem Zustand/Baum s und gegebenen Buchstaben a einen neuen Zustand/Baum s' konstruiert.

1. Färbe jeden grünen Knoten in einen gelben um.
2. Wende auf die Blätter die Potenzmengen-Konstruktion bzgl. Δ an.
3. Spalte die Blätter-Zustands-Beschriftungen in End- und Nichtendzustände auf und erzeuge dafür neue Kinder, links grün, rechts rot. Entferne alte Zustands-Beschriftung.
4. Streiche mehrfach vorkommende Zustände: Behalte am weitesten links stehendes Vorkommen.
5. Entferne alle Knoten, von denen aus nur Blätter erreichbar sind, die mit $\emptyset$ beschriftet sind. (Falls nichts mehr übrig bleibt, d. h. der leere Baum, wird die Berechnung nicht fortgesetzt)
6. Ziehe jeden Knoten, der genau einen Nachfolger hat, mit diesem zusammen:

   - der Nachfolger verschwindet
   - Farbe des Knotens:
     - grün, falls Nachfolger gelb oder grün
     - bleibt, sonst

   [Iteriere dies, bis keine Knoten mehr zusammen gezogen werden können.]

Rabin-Bedingung $\Omega = \{(L_{v_0}, U_{v_0}), \dots, (L_{v_{r-1}}, U_{v_{r-1}})\}$ mit $N = \{v_0, \dots, v_{r-1}\}$ und

$$L_{v_i} = \{\text{Zustand } s \mid v_i \text{ gehört nicht zu } s\}$$
$$U_{v_i} = \{\text{Zustand } s \mid v_i \text{ gehört zu } s \text{ und ist grün}\}$$

Bemerkung 43. Die Anzahl der Zustände des deterministischen Rabin-Automaten ist $2^{\mathcal{O}(n \log n)}$. Frage: Gilt dies auch bei deterministischen Muller-Automaten? Dies ist bisher unklar, evtl. reichen $2^{\mathcal{O}(n)}$.

Folgerung 44. *Die von Büchi-Automaten erkannten Sprachen (ω-reguläre Sprachen, oder reguläre ω-Sprachen) sind unter Komplementbildung abgeschlossen.*

Beweis. Aus gegebenen Büchi-Automaten $\mathbb{A}$ konstruiere äquivalenten *deterministischen* Muller-Automat $(A, Q, q_I, \delta, \mathcal{F})$. Betrachte dann den Automaten $\mathbb{B} = (A, Q, q_I, \delta, \{F \subseteq Q \mid F \notin \mathcal{F}\})$. Dann gilt $L(\mathbb{A}) = A^\omega \setminus L(\mathbb{B})$. Wandle dann noch $\mathbb{B}$ in einen Büchi-Automaten um. □

11 Anwendung

Monadische Theorie der natürlichen Zahlen mit Nachfolger (und $<$).
Logik 1. Stufe besteht aus:

- Signatur Σ von Symbolen für Konstanten, Funktionen und Relationen
- Variablen
- Termen: aufgebaut aus Variablen und Symbolen für Konstanten unter Verwendung der Symbole für Funktionen
- atomare Formeln: $t = t'$ für Terme t, t', $R(t_0, \dots, t_{n-1})$ mit R n-stelliges Relationssymbol
- Formeln: aus atomaren Formeln unter Verwendung von $\wedge, \vee, \neg$ und den Quantoren $\exists x, \forall x$
- Interpretation: Variablen werden durch Elemente des Grundbereichs interpretiert.

Monadische Logik 2. Stufe besteht zusätzlich aus:

- Mengen von Variablen für Mengen von Elementen des Grundbereiches, $X, Y, \dots$

- atomare Formel: Xt, X Variable für Menge, t Term. Statt Xt schreiben wir auch $t \in X$.
- Quantoren für Mengen-Variablen: $\exists X$, $\forall X$
- Interpretation: Belegung der Mengen-Variablen durch Mengen von Elementen des Grundbereichs

Hier: eine Struktur $\mathfrak{M} = (\mathbb{N}, \mathrm{suc})$ mit $\mathrm{suc} = \{(i, i+1) \mid i \in \mathbb{N}\}$.

$$\mathrm{Th}_M(\mathfrak{M}) = \{\text{Menge aller mon. Formeln } \varphi \text{ ohne freie Var.} \mid \mathfrak{M} \vDash \varphi\}$$

Satz 45. $\mathrm{Th}_M(\mathfrak{M})$ *ist entscheidbar.*

Hintergrund:

- $\mathrm{Th}_1(\mathbb{N}, +, \cdot)$ ist nicht entscheidbar.
- $\mathrm{Th}_1(\mathbb{N}, +)$ ist entscheidbar.
- $\mathrm{Th}_1(\mathbb{N}, \cdot)$ ist entscheidbar.
- $\mathrm{Th}_1(\mathbb{R}, +, \cdot)$ ist entscheidbar.

Beispiel 46. (a) „$x = 0$“:

$$\neg\exists y(\mathrm{suc}(y, x)) \text{ oder } \forall y \neg\, \mathrm{suc}(y, x)$$

(b) „$x < y$“:

$$\forall X(\underbrace{(x \in X \wedge \forall z \forall z'(z \in X \wedge \mathrm{suc}(z, z') \rightarrow z' \in X))}_{\text{„}X\text{ enthält }x\text{ und alle Nachfahren von }x\text{“}} \rightarrow y \in X) \wedge \neg y = x)$$

(c) „$X = Y$“:

$$\forall x(x \in X \leftrightarrow x \in Y)$$

Korrespondenz zw. Modellen von Formeln und Mengen von ω-Wörtern

Zur Variablenbelegung $\beta\colon \{X_0, \ldots, X_{n-1}\} \rightarrow 2^{\mathbb{N}}$ betrachte ω-Worte u_β über $\{0,1\}^n$ mit

$$u_\beta(i) = (a_0, \ldots, a_{n-1})^T \text{ mit } a_j = \begin{cases} 1 & i \in \beta(X_j) \\ 0 & \text{sonst} \end{cases}, i \in \mathbb{N}.$$

Zum Wort $u \in (\{0,1\}^n)^\omega$ betrachte $\beta_u\colon \{X_0,\dots,X_{n-1}\} \to 2^{\mathbb{N}}$ mit

$$\beta_u(X_j) = \{i \mid u(i) \text{ hat Eintrag 1 in Zeile } j\} = \{i \mid [u(i)]_j = 1\}\,.$$

Sei φ eine monadische Formel mit freien Variablen unter $X_0,\dots,X_{n-1}$. Dann sei

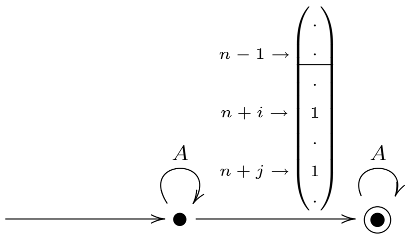
$$L_n(\varphi) = \{u \in (\{0,1\}^n)^\omega \mid \mathfrak{M}, \beta_u \vDash \varphi\}\,.$$

Satz 47. *Für jede monadische Formel φ ist $L_n(\varphi)$ eine reguläre ω-Sprache.*

Beweis. Durch Induktion über den Aufbau der Formel.
Problem: Zwischenformeln können auch Erste-Stufe-Variablen als freie Variablen haben.
Idee: Kodiere Position i bzw. Belegung von x_j durch i durch die Einermenge $\{i\}$. Die Belegung $\beta\colon \{X_0,\dots,X_{n-1}\} \cup \{y_0,\dots,y_{m-1}\} \to 2^{\mathbb{N}} \cup \mathbb{N}$ wird also zu einer Belegung $\beta'\colon \{X_0,\dots,X_{n-1},Y_0,\dots,Y_{m-1}\} \to 2^{\mathbb{N}}$ mit $\beta'(X_i) = \beta(X_i)$ und $\beta'(Y_i) = \{\beta(y_i)\}$.
Zu einer Formel $\varphi = \varphi(X_0,\dots,X_{n-1},y_0,\dots,y_{m-1})$ sei $L_{n,m}(\varphi) = \{u_{\beta'} \mid \mathfrak{M}, \beta \vDash \varphi\}$.
Die Induktionsbehauptung ist dann, für jedes $\varphi = \varphi(X_0,\dots,X_{n-1},y_0,\dots,y_{m-1})$ gilt, $L_{n,m}(\varphi)$ ist eine reguläre ω-Sprache.
Sei $A = \{0,1\}^{n+m}$ und $L_{n,m} = \{u_{\beta'} \mid \beta \text{ ist eine Belegung}\}$. Man kann zeigen, dass $L_{n,m}$ ω-regulär ist.

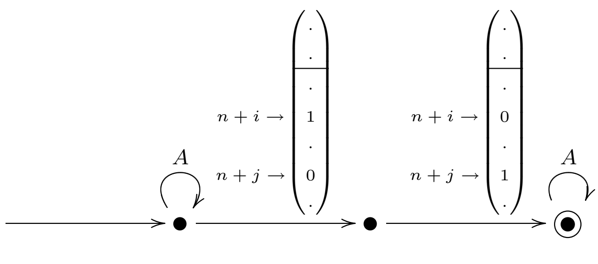
IA: Wir zeigen die Induktionsbehauptung für atomare Formeln:

- $\varphi = (y_i = y_j)$, Œ $i < j$: Sei $\mathbb{A}$ folgender Büchi-Automat:



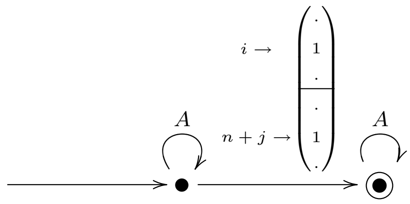
  Dann gilt $L_{n,m}(\varphi) = L(\mathbb{A}) \cap L_{n,m}$, also ω-regulär.

- $\varphi = \mathrm{suc}(y_i, y_j)$, Œ $i < j$: Sei $\mathbb{B}$ folgender Büchi-Automat:



Dann gilt $L_{n,m}(\varphi) = L(\mathbb{B}) \cap L_{n,m}$, also ω-regulär.

- $\varphi = X_i y_j$ $(= y_j \in X_i)$: Sei $\mathbb{C}$ folgender Büchi-Automat:



Dann gilt $L_{n,m}(\varphi) = L(\mathbb{C}) \cap L_{n,m}$, also ω-regulär.

IS: Alle Formeln können so formuliert werden, dass nur die Symbole $\neg$, $\vee$, $\exists X_i$ und $\exists y_j$ verwendet werden.

- $\varphi = \neg\psi$: Dann gilt

$$L_{n,m}(\varphi) = L_{n,m}(\neg\psi) = \underbrace{(A^\omega \setminus \overbrace{L_{n,m}(\psi)}^{\text{reg. nach IV}})}_{\text{reg. nach Folg. 44}} \cap L_{n,m}.$$

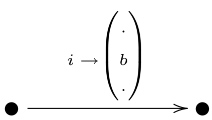
Also ist $L_{n,m}(\varphi)$ ω-regulär.

- $\varphi = \psi_1 \vee \psi_2$: Dann gilt

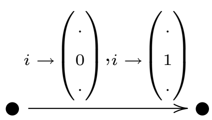
$$L_{n,m}(\varphi) = L_{n,m}(\psi_1 \vee \psi_2) = \underbrace{L_{n,m}(\psi_1)}_{\text{reg. nach IV}} \cup \underbrace{L_{n,m}(\psi_2)}_{\text{reg. nach IV}}$$

Also ist $L_{n,m}(\varphi)$ ω-regulär.

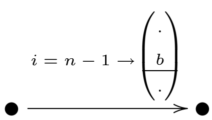
- $\varphi = \exists X_i \psi$: Sei $\mathbb{A}$ der Büchi-Automat für $L_{n,m}(\psi)$. Falls $i < n-1$, erhält man den Automaten für $L_{n,m}(\varphi)$ durch Ersetzen von jeder Transition



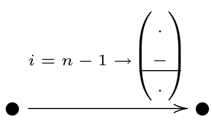
durch



Falls $i = n-1$, so erhält man den Automaten für $L_{n-1,m}(\varphi)$ durch Ersetzen von jeder Transition



durch



Also Streichen der $(n-1)$-ten Zeile in den Transitionen.

- $\varphi = \exists y_j \psi$: Wie $\exists X_i \psi$, zusätzlich $\cap$ mit $L_{n,m}$. Bei $j = m-1$ auch wie oben. $\square$

Satz 48. *Zu jeder regulären ω-Sprache $L \subseteq ((\{0,1\})^n)^\omega$ gibt es eine Formel $\varphi = \varphi(X_0, \dots, X_{n-1})$ mit $L_n(\varphi) = L$.*

Beweis. Sei $\mathbb{A} = (\{0,1\}^n, \{s_0, \dots, s_{m-1}\}, s_0, \Delta, F)$ ein Büchi-Automat, der L erkennt. Definiere φ wie folgt:

$$\begin{aligned}\varphi :=& \exists Y_0 \dots \exists Y_{m-1} \Big\{ \text{„}Y_i \text{ bilden Partition“} \wedge \\ & \text{„}0 \in Y_0\text{“} \wedge \\ & \forall x \forall y \Big[\operatorname{suc}(x,y) \to \bigvee_{\substack{i,j,a:\\ (s_i,a,s_j)\in\Delta}} \Big(Y_i x \wedge Y_j y \wedge \bigwedge_{\substack{i<n\\ a_i=1}} X_i x \wedge \bigwedge_{\substack{i<n\\ a_i=0}} \neg X_i x \Big) \Big] \wedge \\ & \forall x \exists y \Big(\text{„}x<y\text{“} \wedge \bigvee_{i:s_i\in F} Y_i y \Big) \Big\}\end{aligned}$$

Wir müssen noch die Formeln zu den Ausdrücken „Y_i bilden Partition“, „$0 \in Y_0$“ und „$x < y$“ angeben:

- „$x < y$“: siehe Beispiel 46 (b).
- „$0 \in Y_0$“: $\exists x \forall y (\neg \operatorname{suc}(y,x) \wedge Y_0 x)$.
- „Y_i bilden Partition“:

$$\forall x \Big[\bigvee_{i=0}^{m-1} Y_i x \wedge \bigwedge_{i,j:i<j} (Y_i x \to \neg Y_j x) \Big].$$

Intuition: Die Y_i geben an, in welchem Zustand der Automat ist. $j \in Y_i$ bedeutet, der Automat ist nach dem j-ten Schritt im Zustand s_i. $\square$

Folgerung 49. *Jede monadische Formel $\varphi = \varphi(X_0, \dots, X_{n-1})$ ist äquivalent zu einer* existentiellen (universellen) *monadischen Formel, d. h., von der Form $\exists Y_0 \dots \exists Y_{m-1} \psi$ ($\forall Y_0 \dots \forall Y_{m-1} \psi$) mit ψ ohne Mengen-Quantoren.*

Beweis. existentiell: $\varphi \xrightarrow{\text{Satz 47}} L_n(\varphi) \xrightarrow{\text{Satz 48}} \varphi'$.
universell: Komplementbildung und dann wie oben. $\square$

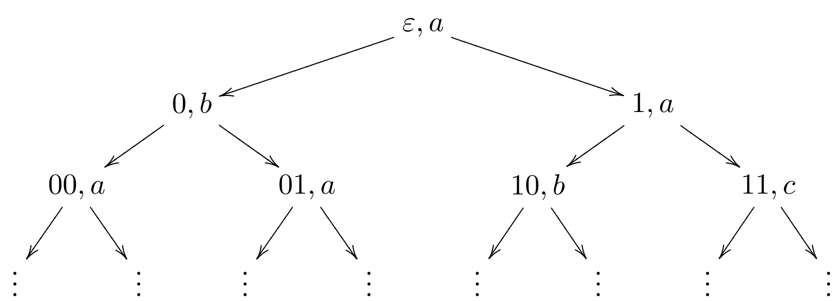
Satz 50. *Die* $\mathrm{Th}_M(\mathbb{N}, \operatorname{suc})$ *ist* nicht *elementar entscheidbar (rekursiv).*

Ohne Beweis.
Eine Sprache L ist *elementar rekursiv* genau dann, wenn $L \in Time(2^{2^{\cdot^{\cdot^{2^n}}}})$ („$\cdots$“ steht für eine feste Höhe). Statt *Time* gilt auch *Space*.

Teil IV

Endliche Automaten auf unendlichen Bäumen

Ein *unendlicher Baum* über dem Alphabet A ist der vollständige Binärbaum mit einer Knotenbeschriftung aus A.



Formal ist ein unendlicher Baum t eine Abbildung $t\colon \{0,1\}^* \to A$. Die *Menge aller Bäume* über A nennen wir T_A. Eine *Baumsprache* über A ist eine Teilmenge $T \subseteq T_A$.

Ein *Muller-Baumautomat* ist ein Tupel $\mathbb{A} = (A, Q, q_I, \Delta, \mathcal{F})$ bestehend aus

- einem Alphabet A,
- einer endlichen Zustandsmenge Q,
- einem Anfangszustand $q_I \in Q$,
- einer Übergangsfunktion $\Delta \subseteq Q \times A \times Q \times Q$ und
- einer Akzeptierbedingung $\mathcal{F} \subseteq 2^Q$.

Analog: Rabin-, Streett-, …Baumautomaten.
Der *Lauf* eines Automaten $\mathbb{A}$ auf einen Baum t ist ein Baum $s \in T_Q$ mit

- $s(\varepsilon) = q_I$ und
- für alle $u \in \{0,1\}^*$ gilt $(s(u), t(u), s(u0), s(u1)) \in \Delta$.

Ein Lauf s heißt *akzeptierend* genau dann, wenn für alle $\alpha \in \{0,1\}^\omega$ das Wort $s|_\alpha = s(\varepsilon)s(\alpha(0))s(\alpha[0,1])s(\alpha[0,2]),\dots$ die Akzeptierbedingung $\mathcal{F}$ erfüllt.

Beispiel 51. (a) Die Sprache $T = \big\{t \in T_{\{a,b\}} \mid \text{es kommt ein } a \text{ vor in } t\big\}$ wird vom Muller-Automaten $(\{a,b\}, \{q_{\text{such}}, q_{\text{ok}}\}, q_{\text{such}}, \Delta, \{\{q_{\text{ok}}\}\})$ mit

Δ	q_{such}	q_{ok}
a	$(q_{\text{ok}}, q_{\text{ok}})$	$(q_{\text{ok}}, q_{\text{ok}})$
b	$(q_{\text{such}}, q_{\text{ok}}), (q_{\text{ok}}, q_{\text{such}})$	$(q_{\text{ok}}, q_{\text{ok}})$

erkannt.

(b) Die Sprache $T = \big\{t \in T_{\{a,b\}} \mid \text{es gibt endlich viele } a \text{ in } t\big\}$ wird vom Muller-Automaten $(\{a,b\}, \{q_a, q_{\text{ok}}\}, q_a, \Delta, \{\{q_{\text{ok}}\}\})$ mit

Δ	q_a	q_{ok}
a	$(q_a, q_a), (q_{\text{ok}}, q_{\text{ok}})$	
b	$(q_a, q_a), (q_{\text{ok}}, q_{\text{ok}})$	$(q_{\text{ok}}, q_{\text{ok}})$

erkannt.

(c) Die Sprache $T = \big\{t \in T_{\{a,b\}} \mid \text{es gibt unendlich viele } a \text{ in } t\big\}$ wird vom Muller-Automaten $(\{a,b\}, \{q_{\text{such}}, q_{\text{ok}}, q_a\}, q_{\text{such}}, \Delta, \{\{q_{\text{ok}}\}, \{q_a\}, \{q_a, q_{\text{such}}\}\})$ mit

Δ	q_s	q_{ok}	q_a

erkannt.

Sei $t\colon \{0,1\}^* \to A$ ein Baum. Ein *unendlicher Pfad* ist $\pi \in \{0,1\}^\omega$. Ein *endlicher* ist $\pi \in \{0,1\}^*$. $t|_\pi$ ist die *Beschriftung* des Pfades $\in A^\infty = A^* \cup A^\omega$. Sei π ein endlicher Pfad, dann ist $t\downarrow_\pi = s$ mit $s(u) = t(\pi u)$.

Lemma 52. *Sei $t \in T_{a,b}$. Dann sind äquivalent:*

(i) $|t^{-1}(a)| = \infty$.

(ii) Es gibt einen unendlichen Pfad π und $i_0 < i_1 < i_2 < \dots$, so dass gilt:

(a) $t(\pi[0,i_j]) = a$ *für alle j* oder

(b) $t_j := t \downarrow \pi[0,i_j](1 - \pi(i_j+1))$ *enthält a für alle j.*

Beweis. (ii.a) $\to$ (i): Klar!

(ii.b) $\to$ (i): Sei δ_j endlicher Pfad derart, dass $t_j(\delta_j) = a$. Dann gilt $t(\pi[0, i_j](1-\pi(i_j+1))\delta_j) = a$. Sei $\rho_j = \pi[0, i_j](1-\pi(i_j+1))\delta_j$. Zeige noch: $\rho_j \neq \rho_{j'}$ für alle $j < j'$. Es gilt aber: $\rho_j = \pi[0, i_j](1-\pi(i_j+1))\dots$ und $\rho_{j'} = \pi[0, i_j]\pi(i_j+1)\dots$, also $\rho_j \neq \rho_{j'}$.

(i) $\to$ (ii): 1. Schritt: Konstruiere unedlichen Pfad π mit folgender Eigenschaft: Für alle i gilt $|(t \downarrow \pi[0,i))^{-1}(a)| = \infty$. Konstruktion per Induktion:

IA $i = 0$: immer erfüllt.

IS $i > 0$: Dann gilt

$$(t \downarrow \pi[0,i))^{-1}(a) = \underbrace{(t \downarrow \pi[0,i)0)^{-1}(a)}_{B_0} \cup \underbrace{(t \downarrow \pi[0,i)1)^{-1}(a)}_{B_1}$$
$$\cup \begin{cases} \emptyset & t(\pi(i)) \neq a \\ \{i\} & \text{sonst} \end{cases}$$

Nach Schubfachprinzip folgt dann aus $|(t \downarrow \pi[0,i))^{-1}(a)| = \infty$ (IV), dass $|B_0|$ oder $|B_1| = \infty$ gilt. Falls $|B_0| = \infty$, setze $\pi(i) = 0$, sonst $\pi(i) = 1$.

2. Schritt: Zeige (b) unter der Annahme, dass (a) nicht gilt. Dazu: Falls (a) nicht gilt, so gibt es ein n mit $\pi(i) \neq a$ für alle $i \geq n$. Konstruiere $i_0 < i_1 < \dots$ induktiv.

IA $k = 0$: Da $|(t \downarrow \pi[0,n])^{-1}(a)| = \infty$ existiert ein u mit $t(\pi[0,i]u) = a$. Sei j maximal mit $\pi[i+1, i+j] = u[0,j)$. Behauptung: $j < |u|$. Falls nämlich $j = |u|$, so $\pi[i+1, i+|u|] = u$, also $\pi[0, i+|u|] = \pi[0,i]u$, d. h., $t(\pi[0, i+|u|]) = a$, Widerspruch. Setze nun $i_0 = i + j$.

IS $k > 0$ Definiere i_k wie i_0, aber ausgehend von $i_{k-1}+1$ anstelle von n. □

Lemma 53 (Königslemma)**.** *Ist $t \subseteq \mathbb{N}^*$ ein gerichteter Baum (präfix abgeschlossen) mit unendlich vielen Knoten und gilt* $\mathrm{Grad}(v) < \infty$ *für jeden Knoten v, so besitzt t einen unendlichen Pfad.*

Ohne Beweis.

12 Abschlusseigenschaften regulärer Baumsprachen

Lemma 54. *Seien $T, T' \subseteq T_A$ reguläre Baumsprachen, so ist auch $T \cup T'$ eine reguläre Baumsprache.*

Beweis. Seien $\mathbb{A} = (A, Q, q_I, \Delta, \mathcal{F})$ und $\mathbb{A}' = (A, Q', q_I', \Delta', \mathcal{F}')$ Baumautomaten mit $T(\mathbb{A}) = T$ und $T(\mathbb{A}') = T'$ und $Q \cap Q' = \emptyset$. Dann gilt für $\mathbb{A}_\cup = (A, Q \cup Q' \cup \{q_N\}, q_N, \Delta \cup \Delta' \cup \{(q_N, a, q_0, q_1) \mid (q_I/q_I', a, q_0, q_1) \in \Delta \cup \Delta'\}, \mathcal{F} \cup \mathcal{F}')$: $T(\mathbb{A}_\cup) = T \cup T'$. $\square$

Ein Spiel $G(\mathbb{A}, t)$ zur Beschreibung, dass ein gegebener Automat $\mathbb{A}$ einen gegebenen Baum t akzeptiert:

- zwei Spieler:
  - Automat (Spieler 0, auch A genannt), wählt Transition
  - Pfadfinder (Spieler 1, auch P genannt), wählt Richtung
- Positionen für Spieler 0 (A): $P_0 = (\pi, q) \in \{0,1\}^* \times Q$
- Positionen für Spieler 1 (P): $P_1 = (\pi, \delta) \in \{0,1\}^* \times \Delta$
- Züge für Spieler 0 (A): $((\pi, q), (\pi, \delta))$, falls $\delta = (q, t(\pi), q_0, q_1) \in Delta$
- Züge für Spieler 1 (P): $((\pi, \delta = (q, a, q_0, q_1)), (\pi b, q_b))$ für $b \in \{0,1\}$
- Anfangsposition: (ε, q_I)
- Pfade sind von der Art $(\varepsilon, q_0)(\varepsilon, \delta_0)(b_0, q_1)(b_0, \delta_1)(b_0 b_1, q_2)(b_0 b_1, \delta_2) \ldots$. Sie sind gewinnbringend für Spieler 0 (A) genau dann, wenn $q_0 q_1 q_2 \ldots$ gewinnbringend bzgl. $\mathcal{F}$ ist.

Lemma 55. *Spieler 0 (A) gewinnt $G(\mathbb{A}, t)$ genau dann, wenn $t \in T(\mathbb{A})$.*

Beweis. „$\Rightarrow$“ Dann besitzt Spieler 0 (A) eine Gewinnstrategie $\sigma \colon P^* P_0 \to P_1$. Definiere einen Lauf s und Baum d über Δ induktiv:

- $s(\varepsilon) = q_I$
- Sei $s(\pi) = q$ definiert mit $|\pi| = n$ und sei $d(\pi[0, n-1))$ definiert, falls $\pi \neq \varepsilon$. Betrachte

$$\begin{aligned} u = &(\varepsilon, s(\varepsilon))(\varepsilon, d(\varepsilon)) \\ &(\pi[0,1), s(\pi[0,1)))(\pi[0,1), d(\pi[0,1))) \\ &(\pi[0,2), s(\pi[0,2)))(\pi[0,2), d(\pi[0,2))) \ldots \\ &(\pi[0,n-1), s(\pi[0,n-1)))(\pi[0,n-1), d(\pi[0,n-1)))(\pi, q). \end{aligned}$$

Beachte: u ist konform mit σ. Sei $\delta = (q, a, q_0, q_1)$ = zweite Komponente von $\sigma(u)$. Setze dann $d(\pi) = \delta$ und $s(\pi b) = q_b$ für $b \in \{0, 1\}$.

„$\Leftarrow$“ Gegeben: Lauf s. Konstruiere σ!

Definiere:

$$\sigma(\dots, (\pi, q)) = (\pi, (s(\pi), t(\pi), s(\pi 0), s(\pi 1))).$$

Zeige noch: Strategie und Lauf sind gewinnbringend bzw. akzeptierend. □

Folgerung 56. *Spieler 1 (P) gewinnt $G(\mathbb{A}, t)$ genau dann, wenn $t \in T_A \setminus T(\mathbb{A})$.*

Beweis. Folgt aus Determiniertheit. □

Idee (für einen Komplementautomat)**.** Der Komplementautomat rät eine Strategie von Spieler 1 (P) und verifiziert sie!
Betrachte das Verifizieren separat:
Gegeben: $\{q_0, \dots, q_{n-1}\}$, Anfangsmenge Q_I, Alphabet B, Transitionsrelation $\Delta \subseteq Q \times B \times Q$ (Œ ginge auch $B = 2^{Q \times Q}$), Akzeptierbedingung $\mathcal{F}$.
Gesucht: *deterministischer* Automat $\mathbb{B} = (B, Q^{\mathbb{B}}, \dots)$ mit $\mathbb{B}$ akz. $b_0 b_1 \dots$ gdw. für alle $q_0 q_1 \dots$ mit $q_0 \in Q_I$, $(q_i, b_i, q_{i+1}) \in \Delta$ gilt $\inf(q_0 q_1 \dots) \notin \mathcal{F}$.

Für alle $q_0 q_1 \dots$ mit $q_0 \in Q_I$, $(q_i, b_i, q_{i+1}) \in \Delta$ gilt $\inf(q_0 q_1 \dots) \notin \mathcal{F}$ gdw. nicht $\exists q_0 q_1 \dots$ mit $q_0 \in Q_I$, $(q_i, b, q_{i+1}) \in \Delta$ und $\inf(q_0 q_1 \dots) \in \mathcal{F}$. D. h. $\mathbb{B}$ erhält man aus $(B, Q, Q_I, \Delta, \mathcal{F})$ durch Komplementierung und anschließender Determinisierung.

Zusätzlich brauchen wir einen Automaten für die Aktualisierung des Gedächtnisses für die Strategie von Spieler 1 (P): $\mathbb{C} = (C, Q^{\mathbb{C}}, q_I^{\mathbb{C}}, \delta^{\mathbb{C}})$ mit $C = \Delta \cup \{0, 1\}$.

$$\mathbb{A} + \mathbb{B} + \mathbb{C} \to \delta$$

mit $\mathbb{A}$ gegebener Baumautomat, $\mathbb{B}$ konstruierter det. Muller-Automat, $\mathbb{C}$ det. endl. Halbautomat und δ der gesuchte Baumautomat.
Zustandsmenge:

$$Q^\delta = \underbrace{\{0,1\}^{Q^{\mathbb{C}} \times \Delta^{\mathbb{A}}}}_{\text{Abb. } Q^{\mathbb{C}} \times \Delta^{\mathbb{A}} \to \{0,1\} \text{geratenene Str. f. Pfadfinder}} \times \underbrace{Q^{\mathbb{B}}}_{\text{Aut. f. Pfadprüfung}} \times \underbrace{2^{Q^{\mathbb{A}} \times Q^{\mathbb{C}}}}_{\text{akt. Zustand des Gedächtnisses}}$$

Akzeptierbedingung: Akzeptierbedingung von $\mathbb{B}$ in der mittleren Komponente.

Transitionsfunktion:

$$((\sigma, s, M), a, (\sigma_0, s_0, M_0), (\sigma_1, s_1, M_1))$$

wobei

- σ_0 und σ_1 geraten werden, d. h. beliebig sind
- für alle $(q, b) \in M$ und alle $\delta = (q, a, q_0, q_1) \in \Delta^{\mathbb{A}}$ wählt Pfadfinder die Richtung $\sigma(b, \delta)$. Formal:

$$T_0 = \big\{((q, b), \delta = (q, a, q_0, q_1)) \in M \times \Delta^{\mathbb{A}} \mid \sigma(b, \delta) = 0\big\}$$

$$T_1 = \big\{((q, b), \delta = (q, a, q_0, q_1)) \in M \times \Delta^{\mathbb{A}} \mid \sigma(b, \delta) = 1\big\}$$

-

$$\begin{aligned} M_0 &= \big\{(q_0, \delta^{\mathbb{C}}(b, (\delta, 0))) \mid ((q, b), \delta = (q, a, q_0, q_1)) \in T_0\big\} \\ &= \big\{(q_0, \delta^{\mathbb{C}}(\delta^{\mathbb{C}}(b, \delta), 0)) \mid ((q, b), \delta = (q, a, q_0, q_1)) \in T_0\big\} \end{aligned}$$

$$\begin{aligned} M_1 &= \big\{(q_1, \delta^{\mathbb{C}}(b, (\delta, 1))) \mid ((q, b), \delta = (q, a, q_0, q_1)) \in T_1\big\} \\ &= \big\{(q_1, \delta^{\mathbb{C}}(\delta^{\mathbb{C}}(b, \delta), 1)) \mid ((q, b), \delta = (q, a, q_0, q_1)) \in T_1\big\} \end{aligned}$$

-

$$\begin{aligned} s_0 &= \delta^{\mathbb{B}}(s, \{(q, q_0) \mid ((q, b), \delta = (q, a, q_0, q_1)) \in T_0\}) \\ s_1 &= \delta^{\mathbb{B}}(s, \{(q, q_1) \mid ((q, b), \delta = (q, a, q_0, q_1)) \in T_1\}) \end{aligned}$$

Fazit. Baumautomaten sind abgeschlossen unter Vereinigung und Komplement, also auch unter Schnitt.

13 Entscheidbarkeit des Leerheitsproblems

Ein *regulärer Baum* ist ein unendlicher Baum, der nur endlich viele Isomorphiklassen von Teilbäumen besitzt. Eine *Repräsentation* $R = (Q, q_I, \delta\colon Q \times \{0, 1\} \to Q, \lambda\colon Q \to A)$ (Q endlich) induziert den regulären Baum $t_R(\pi) = \lambda(\delta^*(q_I, \pi))$.

Satz 57. *Zu jedem Baumautomaten $\mathbb{A}$ lässt sich eine Zahl $n_{\mathbb{A}}$ bestimmen, so dass gilt:*

$$T(\mathbb{A}) \neq \emptyset \text{ gdw. } \exists R(|R| \leq n_{\mathbb{A}} \wedge t_R \in T(\mathbb{A}).$$

Beweis. Idee: Betrachte Spiel zwischen Spieler 0 (Baumkonstruktor) und Spieler 1 (Pfadfinder), so dass Spieler 0 genau dann gewinnt, wenn $T(\mathbb{A}) \neq \emptyset$. Konkret:

- Zustände: $P_0 = Q^{\mathbb{A}}$, $P_1 = \Delta^{\mathbb{A}}$
- Zug: $(q, (q, a, q_0, q_1))$, falls $(q, a, q_0, q_1) \in \Delta^{\mathbb{A}}$ und $((q, a, q_0, q_1), q_{0/1})$
- Gewinnbedingung: Akzeptierbedingung von $\mathbb{A}$ auf P_0.

Behauptung: Spieler 0 gewinnt genau dann, wenn $T(\mathbb{A}) \neq \emptyset$.
Beweis: Wie im Automaten-Pfadfinder-Beweis.
Zur Größe des Baumes: Nach alten Sätzen gibt es auch eine Strategie mit endlichem Gedächtnis (latest appearance record with hit), nach der Spieler 0 gewinnt, wenn er überhaupt gewinnt. Diese lässt sich in einen regulären Baum umsetzen. Sei dazu $\mathbb{C}$ der Gedächtnis-Automat. Dann ist $R = (Q, q_I, \delta, \lambda)$ definiert durch:

$$Q = Q^{\mathbb{A}} \times Q^{\mathbb{C}}$$

$$q_I = \dots$$

$$\delta((q^{\mathbb{A}}, q^{\mathbb{C}}), 0) = (\text{linker Nachfolger in Trans. } \sigma(q^{\mathbb{A}}, q^{\mathbb{C}}), \delta^{\mathbb{C}}(\delta^{\mathbb{C}}(q^{\mathbb{C}}, \sigma(q^{\mathbb{A}}, q^{\mathbb{C}})), 0))$$

$$\lambda((q^{\mathbb{A}}, q^{\mathbb{C}})) = \text{Beschriftung der Trans. } \sigma(q^{\mathbb{A}}, q^{\mathbb{C}})$$

wobei σ die Gewinnstrategie ist. □

Lemma 58. *Es ist entscheidbar, ob ein Baumautomat $\mathbb{A}$ einen regulären Baum t_R gegeben durch R akzeptiert.*

Beweis. Zu gegebenem $\mathbb{A}$ und R konstruiere ein endliches Spiel $G(\mathbb{A}, R)$, so dass gilt: Spieler 0 gewinnt G genau dann, wenn $t_R \in T(\mathbb{A})$.
Ansatz: Konstruiere $G(\mathbb{A}, R)$ wie $G(\mathbb{A}, t)$ (unendliches Spiel), aber geschickt zusammengefaltet.

$$\mathbb{A} = (Q, q_I, \Delta, \mathcal{F}), \quad R = (S, s_I, \delta, \lambda)$$

$$P_0 = S \times Q$$

$$P_1 = S \times \Delta$$

$$((s, q), (s, (q, \lambda(s), q_0, q_1))) \text{ für jedes } (q, \lambda(s), q_0, q_1) \in \Delta \text{ – Züge für Spieler 0}$$

$$((s, (q, a, q_0, q_1)), (\delta(s, b), q_b)) \text{ für jedes } b \in \{0, 1\} \text{ – Züge für Spieler 1}$$

Anfangsposition: (s_I, q_I), Gewinnbedingung: $\mathcal{F}$ auf der zweiten Komponente, nur jede zweite Position. □

Folgerung 59. *Das Leerheitsproblem für Baumautomaten ist entscheidbar.*

14 Monadische Logik zweiter Stufe auf Bäumen: S2S

S2S steht für Second-Order mit zwei Sucessor-Funktionen.
Die monadische Logik für die Struktur $\mathcal{B} = (\{0,1\}^*, \mathrm{suc}_0, \mathrm{suc}_1)$ mit $\mathrm{suc}_0 = \{(u, u0) \mid u \in \{0,1\}^*\}$, $\mathrm{suc}_1 = \{(u, u1) \mid u \in \{0,1\}^*\}$.

Satz 60 (Rabin)**.** $\mathrm{Th}_M(\mathcal{B})$ *ist entscheidbar.*

Ansatz: alles wie bei $(\mathbb{N}, \mathrm{suc})$, siehe Abschnitt 11.
Zu Variablenbelegung (β, γ) mit $\beta \colon \{X_0, \ldots, X_{m-1}\} \to 2^{\{0,1\}^*}$ und $\gamma \colon \{x_0, \ldots, x_{n-1}\} \to \{0,1\}^*$ betrachte Baum $t_\beta^\gamma \colon u \mapsto (v, w) \in \{0,1\}^m \times \{0,1\}^n$ mit $v(i) = 1$ gdw. $u \in \beta(X_i)$ und $w(i) = 1$ gdw. $\gamma(x_i) = u$.
Zu $\varphi = \varphi(X_0, \ldots, X_{m-1}, x_0, \ldots, x_{n-1})$ betrachte $T_m^n(\varphi) = \left\{t_\beta^\gamma \mid (\beta, \gamma) \vDash \varphi\right\}$.
Bemerkung: Ist φ eine Aussage, so gilt: $\mathcal{B} \vDash \varphi (\Leftrightarrow \varphi \in \mathrm{Th}_M(\mathcal{B})) \Leftrightarrow t \in T_0^0(\varphi)$(also $t =$ Binärbaum über $((), ())) \Leftrightarrow T_0^0(\varphi) \neq \emptyset$.

Satz 61. *Zu jedem* $\varphi = \varphi(X_0, \ldots, X_{m-1}, x_0, \ldots, x_{n-1})$ *lässt sich* $\mathbb{A}$ *mit* $T(\mathbb{A}) = T_m^n(\varphi)$ *effektiv konstruieren.*

Beweis. Induktion über φ.
atomare Formeln:

- $\mathrm{suc}_{0/1}(x_i, x_i)$: die Formel ist nicht erfüllbar. Jeder Automat, der $\emptyset$ erkennt, tut's.

- $\mathrm{suc}_b(x_i, x_j)$, $i \neq j$, $b \in \{0,1\}$: Automat entsteht durch Durchschnittbildung eines Automaten $\mathbb{A}$ und des Automaten $\mathbb{A}_m^n$ mit

$$\begin{aligned}\mathbb{A} = (\{q_I, q_t\}, q_I, \{&(q_I, (v,w), q_t, q_I) \mid w(i) = 1, b = 0\} \\ \cup \{&(q_I, (v,w), q_I, q_t) \mid w(i) = 1, b = 1\} \\ \cup \{&(q_I, (v,w), q_I, q_I) \mid w(i) = 0\} \\ \cup \{&(q_t, (v,w), q_I, q_I) \mid w(j) = 1\}, 2^{\{q_I, q_t\}})\end{aligned}$$

  und

$$\mathbb{A}_m^n = \bigcap_{i=0}^{n-1} B_m^i$$

  mit

$$\begin{aligned}B_m^i = (\{q_I, q_{\mathrm{ok}}\}, q_I, \{&(q_I, (v,w), q_{\mathrm{ok}}, q_{\mathrm{ok}}) \mid w(i) = 1\} \\ \cup \{&(q_I, (v,w), q_I, q_{\mathrm{ok}}) \mid w(i) = 0\} \\ \cup \{&(q_I, (v,w), q_{\mathrm{ok}}, q_I) \mid w(i) = 0\} \\ \cup \{&(q_{\mathrm{ok}}, (v,w), q_{\mathrm{ok}}, q_{\mathrm{ok}}) \mid w(i) = 0\}, \{\{q_{\mathrm{ok}}\}\}).\end{aligned}$$

- $x_i = x_i$: $\mathbb{A}^n_m$.
- $x_i = x_i$, $i \neq j$: Durchschnitt von $\mathbb{B}$ und $\mathbb{A}^n_m$ mit

$$\mathbb{B} = (\{q\}, q, \{(q,(v,w),q,q) \mid w(i) = w(j)\}, \{\{q\}\}).$$

- $x_i \in X_j$ ($X_j x_i$): Durchschnitt von $\mathbb{C}$ und $\mathbb{A}^n_m$ mit

$$\mathbb{C} = (\{q\}, q, \{(q,(v,w),q,q) \mid (w(i) = 1) \to (v(j) = 1)\}, \{\{q\}\}).$$

Induktionsschritt:

- $\varphi = \neg\psi$: Sei $\mathbb{A}^n_m(\psi)$ Automat für $T^n_m(\psi)$. Dann ensteht $\mathbb{A}^n_m(\varphi)$ aus dem Komplementautomaten für $\mathbb{A}^n_m(\psi)$ und A^n_m durch Durchschnittsbildung.
- $\varphi = \psi_0 \vee \psi_1$: $\mathbb{A}^n_m(\varphi) =$ „Vereinigung“ von $\mathbb{A}^n_m(\psi_0)$ und $\mathbb{A}^n_m(\psi_1)$.
- $\varphi = \exists X_i \psi$: Sei $\mathbb{A}^n_m(\psi) = (Q, q_I, \Delta, \mathcal{F})$. Dann ist

  $$\mathbb{A}^n_m(\varphi) = (Q, q_I, \{(q,(v,w),q_0,q_1) \mid \exists b((q,(v[i/b],w),q_0,q_1) \in \Delta)\}, \mathcal{F}).$$[3]

  Der Fall $i = m-1$ ist ein Problem, wenn in ψ X_i frei vorkommt, denn: Per IV erhält man nur $\mathbb{A}^n_{m'}(\psi)$ für $m' \geq m$, Induktionsbehauptung fordert aber $\mathbb{A}^n_{m-1}(\varphi)$.

  $$\mathbb{A}^n_m(\varphi) = (Q, q_I, \left\{(q,(v|_{\{0,\dots,m-2\}},w),q_0,q_1) \mid (q,(v,w),q_0,q_1) \in \Delta\right\}, \mathcal{F})$$

- $\varphi = \exists x_i \psi$: Analog und zusätzlich mit $\mathbb{A}^n_m$ schneiden. $\square$

Damit ist der Satz von Rabin (Satz 60) bewiesen.

Beispiel 62. (a) Suche $\varphi(X_0)$ derart, dass $T^0_1(\varphi) =$ Menge aller Bäume t mit $t(u) = (1)$ für ein u (gdw. $\beta \vDash \varphi \Leftrightarrow \exists u(u \in \beta(X_0))$). „Es gibt mindestens eine 1 im Baum.“

$$\exists x_0(X_0 x_0)$$

(b) „x_1 ist Nachfahre von x_0“:

$$\begin{aligned}\varphi_{\leq}(x_0,x_1) = &\forall X_0[X_0x_0 \wedge \forall x_2(X_0x_2 \to \exists x_3\exists x_4(X_0x_3 \wedge X_0x_4 \wedge \\ &\mathrm{suc}_0(x_2,x_3) \wedge \mathrm{suc}_1(x_2,x_4))) \to X_0x_1]\end{aligned}$$

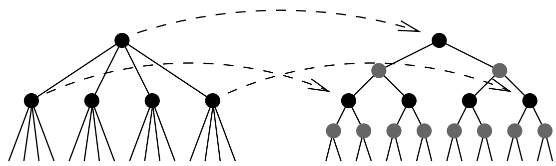
[3] $v[i/b]$ bedeutet, dass der i-te Eintrag in v durch b ersetzt wird

(c) „X_0 ist Antikette“:

$$\varphi(X_0) = \forall x_0 \forall x_1 (X_0 x_0 \wedge X_0 x_1 \wedge \neg x_0 = x_1 \rightarrow (\neg \varphi_{\leq}(x_0, x_1) \wedge \neg \varphi_{\leq}(x_1, x_0)))$$

Satz 63. $\mathrm{Th}_M(\{0, \ldots, k-1\}^*, \mathrm{suc}_0, \ldots, \mathrm{suc}_{k-1})$ *ist entscheidbar.*

Beispiel 64. $k = 4$:



$\mathrm{suc}_2(x_0, x_1) \rightsquigarrow \exists x_2(\mathrm{suc}_1(x_0, x_2) \wedge \mathrm{suc}_0(x_2, x_1))$
$\exists x_i \rightsquigarrow \exists x_i$ „auf gerader Ebene“ $= \exists x_i(\exists X_0(X_0 \ldots) \wedge \ldots)$

Lemma 65. $\varphi(x_0, x_1) \Leftrightarrow |x_0| = |x_1|$ *(x_0 und x_1 haben dieselbe Tiefe) ist nicht definierbar in S2S.*

Beweis. Zeige, dass $T_0^2(\varphi)$ nicht durch einen Baumautomaten erkannt wird. □

Folgerung 66. *Die folgenden Theorien sind entscheidbar:*

- *SkS – Monadische Theorie von k Nachfolgern*
- *SωS – Monadische Theorie von ω vielen Nachfolgern,* $\mathrm{suc}_0, \mathrm{suc}_1, \ldots$.
- *SωB – Monadische Theorie von ω vielen Nachfolgern,* suc.
- *Zweite-Stufe-Theorie der abzählbaren linearen Ordnungen.*
- *Schwache monadische Theorie beliebiger linearer Ordnungen, $\exists X$ endlich, $\forall Y$ endlich.*
- *Zweite-Stufe-Theorie einer einstelligen Funktion über abzählbarem Grundbereich.*
- *Schwache monadische Theorie einer einstelligen Funktion.*
- *Erste-Stufe-Theorie boolscher Algebren mit Quantifizierung über Ideale.*

Satz 67. *Zu jedem Baumautomaten $\mathbb{A}$ über $\{0,1\}^k$ gibt es eine Formel $\varphi(X_0, \ldots, X_{k-1})$, so dass $T(\mathbb{A}) = T_k^0(\varphi)$ gilt.*

Beweis. Wie bei Wörtern: Man beschreibt die Existenz eines akzeptierenden Laufes! □

Für Informatik interessant:

- Informatik-Systeme durch Transitionssysteme beschreiben.
- Spezifizifikation über Transitionssysteme in einer Logik formulieren.
- Überprüfen, ob Spezifikation auf Inf.-System zutrifft (Trans.-Sys. erfüllt Formel).

zwei Mechanismen, die aus einem Graphen mit entscheidbarer mon. Theorie einen neuen Graphen mit entscheidbarer mon. Theorie machen:

- Abwickeln
- inverse rationale Abbildungen

Abwickeln

Gegeben sei $(V, (E_a)_{a\in A})$ mit V Menge von Knoten, $(E_a)_{a\in A} \subseteq V \times V$ Familie von Kantenmengen.
Der *abgewickelte Graph* ist dann $((A\times V)^*, \left\{\alpha(a, v) \xrightarrow{b} \alpha(a, v)(b, w) \mid (v, w) \in E_b\right\}$
Gegeben $(G, B, (r_b)_{b\in B})$ mit

- $G = (V, (E_a)_{a\in A})$,
- Alphabet B und
- zu jedem $b \in B$ sei r_b ein regulärer Ausdruck über $A \cup \bar{A}$ ($\bar{A} = \{\bar{a} \mid a \in A\}$).

Dann sei $G' = (V, (E_a)_{a\in A} \cup (E_{\bar{a}})_{a\in A})$ wobei $E_{\bar{a}} = \{(u, v) \mid (v, u) \in E_a\}$. Dann ist $G'' = (V, (E_b)_{b\in B})$ mit $(u, v) \in E_b$, falls ex. Pfad in G' von u nach v mit Beschriftung $a \in L(r_b)$.

Beispiel 68. (a) Kellerautomat für Anzahl a = Anzahl b:

(b)

Teil V

Rückblick und Ausblick

In der Vorlesung wurden folgende Ideen betrachtet:

- *Alternierende* endliche Automaten (Kozen und Stockmeyer entwickelten diese Idee zunächst für Turing-Maschinen) sind interessant für die Zusammenhänge von Komplexitätsklassen: Läßt man alternierende Automaten für eine Zeitschranke zu, so erhält man die nächsthöhere Platzschranke und umgekehrt (z.B. AP = PSPACE, AL = P). Dabei hat Stockmeyer auch als erster die Komplexität von SIS gezeigt.

  Erweiterungen dafür sind häufig durch die Informatik motiviert, z.B. durch nebenläufige Systeme, die zu parallelel Automaten führen (D. Harel), oder durch Hierarchie-Konzepte (rufe Automaten innerhalb von Automaten als „Unterprogramme“ auf etc., R. Alur). Die alternierenden Automaten dienen dabei zur Modellierung von offenen Systemen: In einer spezifizierten Komponente muß für alle Umweltumgebungen eine Strategie existieren, dies kann man durch alternierende Strukturen modellieren.

- Bei *Spielen* sind die Paritätsspiele und Algorithmen für diese wichtig (Jurdzinski ist einer von vielen, die daran arbeiten).

  Interessant ist es auch, Spiele auf unendlichen Graphen effektiv zu behandeln (Walukiewicz taucht hier oft auf). Dies ist wichtig, da man Systeme in der Informatik häufig mit endlichen Strukturen nur begrenzt modellieren kann – dann ist eine Repräsentierung eines unendlichen Graphen durch geeignete Formalismen nötig, etwa durch einen Keller-Transitions-Graphen, der die Zustände eines Kellerautomaten als Knoten benutzt, und auf dem dann eine Muller-Bedingung definiert werden kann.

  Wichtig sind auch Spiele in Spezifikationslogiken (wie z.B. ATL, Alternating Time Temporal Logic, oder Game Logic, Rohid Parikh).

- *Omega-Sprachen* gehen auf Büchi zurück, wichtig ist auch Boris Trakhtenbrot, wobei die Komplementbildung solcher Automaten entscheident ist (die hier durch Determinisierung betrachtet wurde). Die betrachtete Konstruktion von Muller und Schupp zur Determinisierung ist zuerst von Safra entwickelt worden.

  Die möglichen bzw. untersuchten Erweiterungen sind sehr umfangreich, zur Klassifizierung werden Topologien benutzt (Borel-Hierarchie).

- Bei den *Baumautomaten* ist die Alternierung die wichtigste Erweiterung, die studiert wird; außerdem beliebige Verzweigung.

  Rabin hat den später als Satz von Rabin benannten Satz ohne Zuhilfenahme der Spiele gezeigt, später haben Gurevich und Harrington die Idee gehabt, die Spiele für den Beweis zu nutzen, wobei die hier vorgestellte Version des Beweises auf Zielonka zurückgeht. Ausgehend vom Satz von Rabin kann man versuchen, die Klasse von Graphen zu bestimmen, die eine entscheidbare monadische Theorie haben (z.B. Caucal-Hierarchie) – und dafür die effiziente Auswertung von monadischen Formeln zu betrachten, hierzu gibt es zahlreiche Untersuchungen (z.B. Courselle).

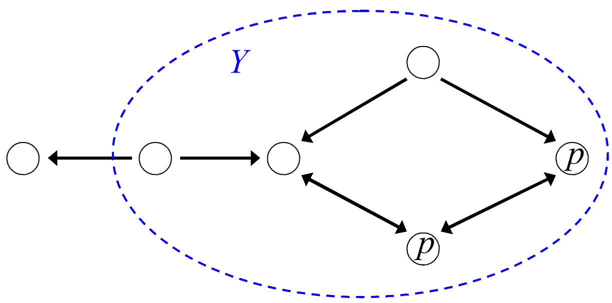
Zudem ist die Komplexität hier eine wichtige Frage – lassen sich Logiken mit gleicher Ausdrucksstärke, aber geringerer Komplexität finden? Am wichtigsten ist hier das modale μ-Kalkül, das von Park und Kozen entwickelt wurde.

14.1 Das modale μ-Kalkül

Das μ-Kalkül wird in Kripkre-Strukturen interpretiert, wir betrachten hier Graphen mit Knotenbewertungen.
Die Formel $Y = \mu X(p \vee \diamond X)$ beschreibt die kleinste Menge X, für die gilt: $X = p \vee \diamond X$, d.h. genauer definiert:

$$X = V_p \cup \{ v \mid \exists w((v,w) \in E \wedge w \in X)\}$$

Dabei bedeutet μ kleinster Fixpunkt, ν größter Fixpunkt, $\diamond$ ein Nachfolger, $\Box$ alle Nachfolger.
Wir werten diese z. B. aus auf dem folgenden Graphen, dessen Knoten mit p oder $\neg p$ beschriftet sind:



Weiteres Beispiel: Die Menge aller Knoten, von denen aus ein unendlicher Pfad existiert, der immer mit p beschriftet ist: $\nu X(p \wedge \diamond X)$.
Der modale μ-Kalkül ist für binäre Bäume genauso ausdrucksstark wie die monadische Logik zweiter Stufe, aber der Kalkül ist sehr viel weniger kompakt. Die Komplexität der Auswertung einer μ-Kalkül-Formel entspricht genau dem Lösen eines Paritätsspiels (es existieren lineare Reduktionen in beide Richtungen). Eine sehr einfache alternative Darstellung sind alternierende Paritäts-Baumautomaten.

Literatur

[1] E. Grädel, W. Thomas, and T. Wilke (eds.). *Automata, Logics, and Infinite Games (A Guide to Current Research).* Volume 2500 of Lecture Notes in Computer Science, pages 61–77, chapter 4. Springer-Verlag, 2002.

[2] M. O. Rabin. *Decidability of second-order theories and automata on infinite trees".* Trans. Amer. Math. Soc. 141 (1969), 1-35.

[3] E. Börger, E. Grädel, and Y. Gurevich. *The classical decision problem.* Springer-Verlag, 1997.