5 Automaten auf unendlichen Bäumen

Unter unendlichen Bäumen versteht man hier solche, die zwar endlichen Verzeigungsgrad haben, aber i.a. unendliche viele Knoten haben und somit auch unendlich lange Äste. Es ist klar, dass sich nur das Top-down-Modell auf unendliche Bäume verallgemeinern lässt.

Man kann dann verschiedene Akzeptanzbedingungen studieren, z.B. Büchi, Muller, Rabin, Parität. Wie bei endlichen Baumautomaten hat diese Akzeptanzbedingung jeweils auf jedem Ast des Eingabebaumes zu gelten. Es zeigt sich dann, dass Automaten mit Büchi-Akzeptanzbedingung echt schwächer sind als die (zueinander äquivalenten) Akzeptanzbedingungen Muller und Parität. Allerdings erreichen auch diese ihre volle Stärke nur in der nichtdeterministischen Version, es gibt hier also keine "Safrakonstruktion". Den Nichtdeterminismus braucht man, um im Baum zu suchen; z.B., ist die Sprache aller Bäume, die einen mit a beschrifteten Knoten enthalten (über $\Sigma = \{a, b\}, \sigma(a) = \sigma(b) = 2$) nicht durch einen deterministischen Muller-Baumautomaten zu erkennen.

Die Büchi-erkennbaren Baumsprachen sind nicht unter Komplement abgeschlossen. Ein konkretes Beispiel für eine Baumsprache, die nicht Büchi-erkennbar ist wie folgt: $\Sigma = \{a, b\}$, $\sigma(a) = \sigma(b) = 2$, $L = \{t \mid \text{auf jedem Pfad finden sich nur endlich viele } b\}$. Man beachte, dass $\overline{L}$ Büchi-erkennbar ist.

Man könnte es etwa mit dem Automaten $\mathcal{A} = (\{q_0, q_1\}, \{a, b\}, q_0, \delta, \{q_1\})$ wobei $\delta_a(q_0) = \delta_b(q_0) = Q^2$ und $\delta_b(q_1) = \emptyset$ und $\delta_a(q_1) = \{(q_1, q_1)\}$, versuchen.

Betrachtet man aber den Baum t mit $t(w) = b \iff w \in 1^+0$, so sieht man leicht, dass $\mathcal{A}$ ihn nicht erkennt, obwohl er zu L gehört. Auf dem Pfad 1^ω muss ja irgendwann der Zustand q_1 kommen und dann kann man nicht mehr die b's verarbeiten, die links von ihm abzweigen. Ein rigoroser Beweis, dass L nicht Büchi-erkennbar ist, verläuft ähnlich.

Ein Paritätsbaumautomat (PBA) für L ist dagegen leicht anzugeben, in diesem Fall sogar deterministisch: $\mathcal{A} = (\{q_0, q_1\}, q_0, \delta, \Omega)$, wobei $\delta_a(q) = \{(q_0, q_0)\}$ und $\delta_b = \{(q_1, q_1)\}$ und $\Omega(q_0) = 0, \Omega(q_1) = 1$. Ein Lauf des PBA ist definitionsgemäß akzeptierend, wenn auf jedem Pfad die größte unendlich oft vorkommende Priorität gerade ist. Vorkommen von a signalisieren wir durch Priorität 0; Vorkommen von b durch die Priorität 1. Diese muss also auf jedem Pfad nach endlicher Zeit aussterben.

Wir werden uns im folgenden auf das Studium der Paritätsbaumautomaten beschränken.

5.1 Unendliche Bäume

Im folgenden sei Σ ein Baumalphabet ohne Blätter, d.h., $\sigma(a) > 0$ für alle $a \in \Sigma$. Das ist eine technische, vereinfachende Bedingung; wenn gewünscht, kann man Blätter durch einstellige Symbole simulieren mit der Maßgabe, dass der Teilbaum unterhalb eines solchen Symbols "nicht gilt".

Definition 43
Ein unendlicher Baum über Σ ist eine nichtleere, präfixabgeschlossene Menge $T \subseteq \mathbb{N}^*$ und eine Funktion $t : T \to \Sigma$, sodass $\forall w \in T. \forall i < \sigma(t(w)). wi \in T$.

Beachte, dass T stets unendlich ist.

Ein Pfad in einem unendlichen Baum t ist ein Wort $\pi \in \mathbb{N}^\omega$ derart dass jedes endliche Präfix u von w in $\mathrm{dom}(t)$ ist. Der Pfad definiert ein unendliches Wort $t(\pi) \in \Sigma^\omega$ durch $t(\pi)_i = t(w_0, \dots, w_{i-1})$.

5.2 Paritätsbaumautomaten

Definition 44
Ein Paritätsbaumautomat (PBA) ist ein Tupel $\mathcal{A} = (Q, \Sigma, q_0, \delta, \Omega)$, wobei Q eine endliche Menge von Zuständen ist, Σ ein Alphabet wie oben, $q_0 \in Q$, $\delta_a : Q \to 2^{Q^{\sigma(a)}}$ für $a \in \Sigma$, $\Omega : Q \to \mathbb{N}$.

Ein Lauf des PBA auf einem unendlichen Baum t ist eine Funktion $r : \mathrm{dom}(t) \to Q$ sodass $r(\epsilon) = q_0$ und $(r(w0), \dots, w(\sigma(t(w)) - 1)) \in \delta_{t(w)}(r(w))$ für alle $w \in \mathrm{dom}(t)$.

Ein Lauf ist akzeptierend, wenn für alle Pfade π in t gilt, dass die größte Zahl, die in der Folge $\Omega(r(\pi))$ vorkommt, gerade ist.

Die von $\mathcal{A}$ erkannte Sprache besteht wie üblich aus allen Bäumen, zu denen ein akzeptierender Lauf existiert. Analog definiert man Läufe und akzeptierende Läufe von anderen Zuständen als q_0

5.2.1 Latest Appearance Records

Wir müssen nun noch ein Lemma nachholen, welches eigentlich zu den Wortautomaten gehört.

Satz 60
Sei $L \subseteq \Sigma^\omega$ eine ω-reguläre Sprache. Dann ist L von einem deterministischen *Paritäts(wort)automaten erkennbar.*

Beweis Mithilfe der Safrakonstruktion können wir annehmen, dass L von einem deterministischen Mullerautomaten $\mathcal{A} = (\Sigma, Q, q_0, \delta, \mathcal{F})$ erkannt wird. Wir konstruieren einen deterministischen Paritätsautomaten $\mathcal{P} = (\Sigma, Q', q_0', \delta', \Omega)$ wie folgt.

Die Zustandsmenge Q' ist gegeben als $Q! \times |Q|$. Ein Zustand ist also ein Paar (l, i), wobei l eine wiederholungsfreie Liste (Permutation) der Zustände von $\mathcal{A}$ ist und i eine Zahl im Bereich $0 \dots |Q| - 1$ ist.

Der Anfangszustand ist eine beliebige Liste, an deren Anfang der Anfangszustand des Mullerautomaten steht. Auch die Positionszahl i ist beliebig.

Die Übergangsfunktion δ' ist wie folgt erklärt:

$$\delta'(((q_0, q_1, ..., q_{n-1}), i), a) = ((q_j, q0, q1, ..., q_{j-1}, qj+1, ...qn-1), j), \text{ falls } \delta(q_0, a) = q_j$$

NB: q_0 bezeichnet hier nicht den Anfangszustand, sondern den ersten Zustand in der Liste.

Man sucht sich also den Folgezustand, der dem ersten Zustand in der Liste entspricht und schreibt diesen nach vorne. Die anderen Zustände rücken entsprechend nach hinten. Den Positionszeiger setzen wir auf die Position, von der der neue Zustand geholt wurde.

Das bedeutet, dass nach einer Einschwingzeit die Permutationskomponente des Zustands immer die Reihenfolge des letzten Vorkommens aller Zustände visualisiert. Man bezeichnet sie deshalb als Latest Appearance Record (LAR).

Es verbleiben noch die Prioritäten. Wir setzen fest:

$$\Omega((q0, q1, ..., qn-1), i) = 2i, \text{ falls } \{q_0, q_1, ..., q_i\} \in \mathcal{F}$$
$$\Omega((q0, q1, ..., qn-1), i) = 2i+1, \text{ falls } \{q_0, q_1, ..., q_i\} \in \mathcal{F}$$

Es verbleibt zu zeigen, dass dieser Automat dieselben Wörter, wie $\mathcal{A}$ erkennt. Sei $w \in L(\mathcal{A})$ und $U = \inf(w) \in \mathcal{F}$. Beim Abarbeiten des Wortes w werden sich ab einem gewissen Zeitpunkt die Zustände aus U am Anfang des LAR befinden und nur noch unter diesen werden Zustände nach vorne geholt, ausserdem wird jeder immer wieder nach vorne geholt. Das bedeutet, dass genau die Positionen $\{0 \ldots, i\}$ immer wieder auftreten; die größeren sterben aus. Die größte Position, die immer wieder auftritt, ist i und bei ihrem Auftreten ist $\{q_0, \ldots, q_i\} = U \in \mathcal{F}$, somit ist die korrespondierende Priorität gerade und der Paritätsautomat akzeptiert.

Sei jetzt w nicht in $L(\mathcal{A})$. Dann ist $\inf(w) \notin \mathcal{F}$ und deshalb mit demselben Argument die größte unendlich oft auftretende Priorität ungerade. Also ist w auch nicht in $L(\mathcal{P})$. ■

Wir bemerken, dass man mit dem LAR auch das DJW Spiel lösen kann. Der Zahlenspiel führt einen LAR mit, in den er immer die gespielten Buchstaben einträgt; er antwortet jeweils mit dem aktuellen Positionszeiger.

Satz 61
Die Klasse der PBA-erkennbaren Baumsprachen ist unter Vereinigung und Projektion entlang stelligkeitserhaltender Funktionen $\Sigma' \to \Sigma$ zwischen Baumalphabeten abgeschlossen.

Beweis Wie im endlichen Falle unter Verwendung von Nichtdeterminismus. ■

Satz 62
Sei Σ Baumalphabet mit maximaler Stelligkeit n und $D = \{0, \ldots, n-1\}$. Sei L eine ω-reguläre Wortsprache über $\Sigma \times D$. Die assoziierte Baumsprache L^t besteht aus allen Bäumen t, sodass jeder Pfad in t geschrieben als Wort über $\Sigma \times D$ in L ist. Die Sprache L^t ist PBA-erkennbare Baumsprache.

Beweis Man bestimt mit Satz 60 einen deterministischen Paritätsautomaten $\mathcal{P} = (\Sigma \times D, Q, q_0, \delta, \Omega)$. Der folgende PBA erkennt offensichtlich L^t:

$$(\Sigma, Q, q_0, \delta', \Omega)$$

wobei

$$\delta'_a(q) = (\delta(q, (a, 0)), \delta(q, (a, 1)), \ldots, \delta(q, (a, n-1)))$$

Der entstehende PBA ist hier also sogar deterministisch. Es ist allerdings leider nicht möglich, dieselbe Konstruktion ausgehend von einem nichtdeterministischen Paritätsautomaten durchzuführen, selbst dann nicht, wenn man in Kauf nimmt, dass der entstehende PBA nichtdeterministisch wird. Der Grund ist, dass dann der Pfadfinder zu stark wird; er kann nämlich den gewählten Pfad von den nichtdeterministischen Entscheidungen des Wortautomaten abhängig machen. Für ein konkretes Gegenbeispiel betrachte man $\Sigma = \{a, b\}, \sigma(a) = \sigma(b) = 2$, $L = (b \times D)^*(a \times D)^\omega$. Hier ist L^t gerade die in der Einführung erwähnte nicht-Büchi-erkennbare Baumsprache. Ein nichtdeterministischer Büchiautomat würde hier einfach raten, wann die a's vorbei sind. Lässt man ihn auf allen Pfaden laufen, dann müsste er aber in der Lage sein, festzustellen, wann die a's auf allen Folgepfaden vorbei sind. Das muss aber möglicherweise nie passieren!

5.3 Paritätsspiele

Wir wollen für die Komplementierung von PBA wiederum ein Spiel zwischen dem Automaten und dem Pfadfinder einsetzen. Dafür ist es nötig, die dabei auftretenden Spiele zunächst separat zu studieren.

Definition 45 (Paritätsspiel)
Ein Paritätsspiel ist ein Tupel $G = (\mathrm{Pos}_A, \mathrm{Pos}_P, \delta, \Omega)$, wobei Pos_A und Pos_P zwei disjunkte, nicht notwendigerweise endliche Mengen sind, die Positionen des Spiels. Man schreibt $\mathrm{Pos} := \mathrm{Pos}_A \cup \mathrm{Pos}_P$. Die Relation $\delta \subseteq \mathrm{Pos} \times \mathrm{Pos}$ modelliert die möglichen Züge und $\Omega : \mathrm{Pos} \to \{0, \dots, n\}$ Positionen Prioritäten aus einer stets endlichen Teilmenge von $\mathbb{N}$ zuweist.

Das Spiel wird von einer gegebenen Startposition $p \in \mathrm{Pos}$ so gespielt, dass eine Figur auf p gesetzt wird und dann von den Spielern A und P entlang der Kanten δ verschoben wird. Ist die aktuelle Position in Pos_A, so ist Spieler A am Zug, ansonsten P.

Kann ein Spieler nicht mehr ziehen, so hat er verloren, ansonsten wird das Spiel ad infinitum fortgesetzt. Die resultierende unendliche Partie $p = p_0, p_1, p_2, p_3, \dots$ wird von Spieler A gewonnen, falls die größte Zahl, die in der Folge $\Omega(p_0), \Omega(p_1), \dots$ unendlich oft auftritt, gerade ist. Ansonsten, also wenn sie ungerade ist, gewinnt P.

Eine Gewinnposition für A ist eine Position, von der aus A das Spiel bei beliebigem Spiel von P gewinnen kann. Die Menge der Gewinnpositionen für A wird mit W_A bezeichnet. Analog definiert man W_P.

Eine positionale Gewinnstrategie für A ist eine Funktion $s : W_A \cap \mathrm{Pos}_A \to W_A$, die für jede Gewinnposition p für A, in der auch A am Zuge ist, einen legalen Zug $s(p)$, also $(p, s(p)) \in \delta$ ausweist, sodass A von jeder beliebigen Gewinnposition aus tatsächlich gewinnt, wenn er nur die von der Strategie s empfohlenen Züge durchführt.

Satz 63 (Positionale Determiniertheit von Paritätsspielen)
Sei $G = (Pos_A, Pos_P, \delta, \Omega)$ ein Paritätsspiel. Es gilt $W_A \cup W_P = Pos$ und es existieren positionale Gewinnstrategien für A und für P.

Beweis Die Tatsache, dass von jeder Position entweder A oder P das Spiel gewinnen kann, wird als Determiniertheit bezeichnet und folgt aus einem allgemeinen Satz (Martins (nicht MH) Determiniertheitssatz). Die Determiniertheit ist nicht offensichtlich und in der Tat folgt aus dem Auswahlaxiom die Existenz unendlicher Spiele, die nicht determiniert sind.

Die positionale Determiniertheit ist eine Besonderheit der Paritätsspiele und gilt z.B. nicht für die in offensichtlicher Weise definierten Mullerspiele, zu denen das DJW-Spiel gehört.

Wir beweisen sie nunmehr direkt ohne Rückgriff auf den Determiniertheitssatz durch Induktion über n, die höchste vorkommende Priorität.

Ist $n = 0$, so gewinnt A bei beliebigem Spiel positional. Sei also $n > 0$. Indem wir ggf. die Rollen von A und P vertauschen, dürfen wir o.B.d.A. voraussetzen, dass n gerade ist.

Wir zeigen zunächst:

Lemma 33
Sei U eine Menge von Positionen, sodass A von jeder Position $p \in U$ eine positionale Gewinnstrategie s_p besitzt. Dann gibt es auch eine einzige Gewinnstrategie s, deren Befolgung Gewinn für A von allen Positionen $p \in U$ sichert.

Beweis Wir nummerieren die Positionen durch: $p_0, p_1, p_2, \dots$ (falls überabzählbar viele, so verwende man Ordinale). Jetzt betrachten wir s_{p_0}. Diese Strategie sichert den Gewinn für A von p_0 aus, aber auch von allen Positionen aus, die bei Befolgung von s_0 von p_0 aus erreichbar sind bei beliebigem Spiel von P, denn was auch immer P macht, muss ja die Gewinnstrategie funktionieren. Natürlich sind das nicht alle Positionen; aber es gibt ja noch s_{p_1}. Dies sichert Gewinn auf p_1 und allen Positionen, die von p_1 erreichbar sind. Somit können wir die universelle Gewinnstrategie s definieren durch $s(p) = s_{p_i}(p)$, wobei i das kleinste i ist, sodass p von p_i aus bei Befolgung von s_{p_i} erreicht werden kann. ■

Sei jetzt U die Menge derjenigen Positionen, von denen aus P eine positionale Gewinnstrategie besitzt. Nach Lemma 33 gibt es dann eine einzige Gewinnstrategie, die P's Gewinn auf allen Positionen in U sichert. Wir möchten zeigen, dass von allen Positionen in $\mathrm{Pos} \setminus U$ der Spieler A eine positionale Gewinnstrategie hat.

Wir betrachten das Teilspiel auf den Positionen außerhalb von U, d.h. alle Spielzüge, die aus $\mathrm{Pos} \setminus U$ herausführen, werden verboten. Wir beachten, dass es keinen so herausführenden Spielzug für P geben kann, da ja sonst die entsprechende Position zu U gegeben hätte werden müssen. Außerdem wird Spieler A durch diese Wegnahme nicht in eine Sackgasse geführt, denn gäbe es eine Position $p \in \mathrm{Pos}_A \setminus U$ derart, dass jeder Zug von p nach U hineinführt, so hätte man diese Position p auch der Menge U hinzurechnen müssen.

Kommt in $\mathrm{Pos} \setminus U$ die höchste Priorität nicht vor, so erlaubt uns die Induktionshypothese die Aufteilung von $\mathrm{Pos} \setminus U$ in zwei Bereiche W_A, W_P, wobei A das Teilspiel in W_A gewinnt und P es in W_P gewinnt (jeweils positional). Nachdem P den Bereich $\mathrm{Pos} \setminus U$ nicht verlassen kann, gewinnt A auf W_A sogar das gesamte Spiel. Andererseits gewinnt P auf W_P auch das gesamte Spiel, denn sollte A sich in den Bereich U absetzen, so würde

P natürlich sofort seine dortige Strategie zum Einsatz bringen. Aufgrund der Annahme an U bedeutet dies aber, dass W_P leer ist und somit A auf ganz Pos $\setminus U$ eine positionale Gewinnstrategie besitzt, w.z.b.w.

Es bleibt der Fall zu betrachten, wo die höchste Priorität n (o.B.d.A. n gerade) in Pos$\setminus U$ vorkommt. Sei N die Menge der Positionen in Pos$\setminus U$ mit Priorität n. Sei $Attr_A(N)$ die Menge derjenigen Positionen, von denen aus A einen Besuch von N erzwingen kann. Man nennt diese Menge den Attraktor von N. Man beachte, dass es keine A-Züge nach $Attr_A(N)$ hinein gibt und keine P-Züge aus $Attr_A(N) \setminus N$ heraus. Es gilt stets $N \subseteq Attr_A(N)$.

Wir betrachten nunmehr das Teilspiel auf $Z = \text{Pos} \setminus U \setminus Attr(N)$. Wieder handelt es sich hierbei tatsächlich um ein Spiel, denn würden von einer gewissen Position alle Züge von P nach $Attr(N)$ hineinführen, so hätte man diese Position dem Attraktor zuschlagen müssen. Wir teilen es nach Induktionshypothese auf in Gewinnbereiche W_A und W_P. Auf W_P hat P sogar für das gesamte Spiel eine positionale Gewinnstrategie, denn in den Attraktor kann A nicht entkommen und ein Entkommensversuch in den Bereich U führt zum Verlust. Daher ist aber wie vorher W_P leer.

Es bleibt zu zeigen, dass sowohl auf dem Attraktor, als auch auf W_A das Gesamtspiel von A positional gewonnen wird. Das geht wie folgt: Auf W_A spielt A gemäß seiner positionalen Gewinnstrategie für das Teilspiel; auf dem Attraktor hingegen erzwingt A einen Besuch von N. Diese Strategie stellt insbesondere sicher, dass der Spielverlauf Pos $\setminus U$ nie verlässt. Führt der Spielverlauf immer wieder nach N hinein, so gewinnt A nach Definition, ansonsten gemäß der Induktionshypothese.

Die positionale Determiniertheit der Paritätsspiele ist somit erwiesen. ■

5.3.1 Algorithmische Behandlung der Paritätsspiele

Ist die Menge der Positionen eines Paritätsspiel endlich, so kann man nach Algorithmen fragen, die entscheiden, ob eine bestimmte Position in W_A oder in W_P ist. Ein einfacher Algorithmus von McNaughton und Zielonka ist in Abbildung 5.1 dargestellt und funktioniert in etwa wie folgt:

Sei N die Menge der Positionen mit der höchsten (o.B.d.A. geraden Priorität). Berechne iterativ den Attraktor $Attr_A(N)$, also die Menge der Positionen von denen aus A das Spiel nach N zwingen kann. Berechne nunmehr rekursiv (die Zahl der Prioritäten ist ja kleiner geworden) die Menge der Positionen X, von denen aus P das Teilspiel Pos $\setminus Attr_A(N)$ gewinnt. Ist diese leer, so gewinnt A überall. Ansonsten gewinnt P auch das gesamte Spiel von allen Positionen in X, da es ja keine Züge für A in den Attraktor gibt. Sei nun Y die Menge der Positionen im Gesamtspiel, von denen aus P den Spielverlauf nach X zwingen kann (auch ein Attraktor). Es ist klar, dass P auch auf allen Positionen in Y gewinnt. Nun bestimmen wir rekursiv W'_A und W'_P für das Teilspiel Pos $\setminus Y$ (weniger Positionen!). Wir behaupten, dass $W_A = W'_A$ und $W_P = Y \cup W'_P$. In der Tat gewinnt A von W'_A aus, da ja P nicht in den Attraktor Y entkommen kann. Auf der anderen Seite gewinnt P von W'_P aus, da ein mögliches Entkommen von A in die Menge Y sofort mit der dort vorhandenen Gewinnstrategie beantwortet wird.

Abb. 5.1 zeigt diesen Algorithmus in Pseudocode. Dabei ist $G \setminus N$ für ein Spiel G

```
SolveGame(G) =
    let (Pos_A, Pos_P, δ, Ω) := G
    n := max{Ω(v) | v ∈ Pos}

    if  n = 0 then return (W_A := Pos, W_P := ∅)

    if n gerade then σ := A else σ := P

    N := {v ∈ Pos | Ω(v) = n}
    N' := Attr_σ(N)

    (W'_σ, W'_σ̄) := SolveGame(G \ N')
    if W'_σ̄ = ∅ then return (W_σ := W'_σ ∪ N', W_σ̄ := ∅)

    N'' := Attr_σ̄(W'_σ̄)
    (W''_σ, W''_σ̄) := SolveGame(G \ N'')
    return (W_σ := W''_σ, W_σ̄ := N'' ∪ W''_σ̄)
```

Abbildung 5.1: Ein rekursiver Algorithmus zum Lösen von Paritätsspielen.

und eine Menge N von Positionen darin das Spiel, welches aus G entsteht, wenn alle Positionen in N mit allen ein- und ausgehenden Kanten entfernt werden.

Leider werden in diesem Algorithmus zwei rekursive Aufrufe getätigt, sodass die Komplexität exponentiell ist. Es ist ein offenes Problem, ob sich Paritätsspiele in polynomialer Zeit entscheiden lassen.

Satz 64
Das Problem, Paritätsspiele zu lösen, ist in NP∩co-NP.

Beweis Es reicht, sich auf das etwas speziellere Problem zu beschränken: Gegeben ist ein Paritätsspiel $G = (Pos_A, Pos_P, \delta, \Omega)$ und ein $v \in Pos$. Entscheide, ob $v \in W_A$ ist. Da es nur $|Pos|$ viele Positionen gibt, kann man einen Algorithmus für dieses Problem leicht in einen Algorithmus für das allgemeine Paritätsspielproblem umwandeln, dessen Laufzeit lediglich um den Faktor $|Pos|$ höher ist.

Die Inklusion in NP wird wie folgt gezeigt. Rate eine positionale Strategie für Spieler A, die den Knoten v enthält. Das ist offensichtlich in Zeit $O(|\delta|)$ möglich. Betrachte nun den von dieser Strategie s induzierten Graphen G'. Dieser enthält nur noch die Kanten

$$\{(w, s(w)) \mid w \in Pos_A\} \cup (\delta \cap Pos_P \times Pos)$$

Auch diese Konstruktion ist in Zeit $O(|\delta|)$ möglich.

Nun gilt: s ist eine Gewinnstrategie gdw. die höchste Priorität auf jedem in G' vorkommenden Zykel gerade ist. Diese Bedingung ist aber ebenfalls in polynomieller Zeit prüfbar.

Die Inklusion in co-NP folgt aus der Determiniertheit. Beachte: Ein Problem ist in co-NP, falls sein Komplement in NP ist. D.h. für die Inklusion in co-NP muss man einen NP-Algorithmus angeben, der entscheidet, ob $v \notin W_A$ ist. Dies ist zuerst nicht offensichtlich, denn dazu müsste man ja eigentlich jede Strategie testen. Es gilt aber: $v \notin W_A$ gdw. $v \in W_P$. Damit lässt sich obiger NP-Algorithmus genauso verwenden, indem lediglich nach einer Strategie für Spieler P gesucht wird.

5.4 Komplementierung der Paritätsbaumautomaten

Die schwierigste logische Abschlusseigenschaft ist der Abschluss der PBA unter Komplement. Wie schon bei endlichen TDBA fassen wir die Akzeptanz durch den Automaten als Zweipersonenspiel auf zwischen Spieler A (Automat) und P (Pfadfinder). Sei also ein PBA $\mathcal{A} = (Q, \Sigma, q_0, \delta, \Omega)$ und ein Baum t gegeben. Das assoziierte Paritätsspiel $G(\mathcal{A}, t)$ ist wie folgt erklärt:

1. Positionen des Spiels sind Paare (w, q) und Paare $(w, \vec{q})$, wobei $w \in D^*$ eine Position im Baum ist ($D = \{0, \dots, n-1\}$, q ein Zustand und $\vec{q} \in Q^{<n}$ ein Tupel von Zuständen ist (n ist die maximale Stelligkeit des Alphabets.)

   Bei Positionen der Form (w, q) ist A am Zug; bei Positionen der Form $(w, \vec{q})$ ist P am Zug.

2. In der Position (w, q) wählt A ein Tupel $\vec{q} \in \delta_a(q)$, wobei $a = t(w)$ und erreicht die Position $(w, \vec{q})$).

3. In der Position $(w, \vec{q})$ wählt P eine Richtung $i < \sigma(t(w))$ und es wird die Position (wi, q_i), wobei q_i die i-te Komponente von $\vec{q}$ ist.

4. Die Priorisierung ist gegeben durch $\Omega(w, q) = \Omega(q)$ und $\Omega(w, \vec{q}) = 0$.

Lemma 34
Spieler A gewinnt das Spiel $G(\mathcal{A}, t)$ von der Position (ϵ, q_0) aus, genau dann, wenn $t \in L(\mathcal{A})$.

Beweis Übung. ∎

Mit der positionalen Determiniertheit der Paritätsspiele folgt hieraus, dass $t \notin L(\mathcal{A})$ genau dann, wenn P das Spiel von (ϵ, q_0) aus positional gewinnt. Dies nutzen wir nun, um einen PBA für das Komplement von $L(\mathcal{A})$ zu konstruieren.

Satz 65 (Komplementierung von PBA)
Sei $\mathcal{A} = (Q, \Sigma, q_0, \delta, \Omega)$ PBA. Man kann effektiv einen PBA $\mathcal{A}'$ konstruieren, mit $L(\mathcal{A}') = \overline{L}(\mathcal{A})$.

Beweis Wenn $t \notin L(\mathcal{A})$ ist, so besitzt P im Spiel $G(\mathcal{A}, t)$ eine positionale Gewinnstrategie. Diese hat die Form einer Funktion

$$s : \mathrm{dom}(t) \times \Sigma \times Q^n \to D$$

wobei n die maximale Stelligkeit von Σ ist und $D = \{0, \dots, n-1\}$.

Wir setzen $S = \Sigma \times Q^n \to D$, sodass die Funktion s als mit S beschrifteter Baum aufgefasst werden kann.

Sei $\Delta = \Sigma \times S, p_1(a, s) = a, p_2(a, s) = s$.

Es ist also $t \notin L(\mathcal{A})$ genau dann, wenn ein Baum t' (über Δ) existiert, derart, dass $t := p_1(t')$ und $s := p_2(t')$ eine Gewinnstrategie in $G(\mathcal{A}, t)$ für P definiert.

Sei L' die Menge aller solchen Bäume t'.

Ein Baum t' ist in L' genau dann, wenn für alle Zugfolgen $z = \vec{q}_0, \vec{q}_1, \vec{q}_2, \dots$ für A folgendes gilt: Die durch w und $s = p_2(t')$ definierte Partie ist entweder illegal und der Fehler dafür liegt bei w, also A, oder sie ist ein Gewinn für P.

Da jede Partie einen Pfad durch t definiert, können wir die Partien auch nach diesen Pfaden aufteilen:

Ein Baum t' ist in L' genau dann, wenn für alle Pfade π und alle Zugfolgen $z = \vec{q}_0, \vec{q}_1, \vec{q}_2, \dots$ für A folgendes gilt:

Die durch w und $s = p_2(t')$ definierte Partie ist entweder illegal und der Fehler dafür liegt bei w, also A und falls die Partie den Pfad π definiert, so ist sie ein Gewinn für P. Dies aber lässt sich als MSO definierbare und damit ω-reguläre Eigenschaft F von Pfaden in t' formalisieren; somit ist $L' = F^t$ und nach Satz 62 durch einen PBA erkennbar. Den gewünschten Komplementautomaten erhält man dann durch nichtdeterministisches Raten der Strategiebeschriftung gemäß Satz 61. ∎

5.5 Leerheitstest auf Baumautomaten

Definition 46
Sei $\mathcal{A} = (Q, \Sigma, q_0, \delta, \Omega)$ ein PBA, und $m := \max\{\sigma(a) \mid a \in \Sigma\}$. Sei $\Sigma^- := \{a\}$ mit $\sigma(a) := m$.

Definiere einen PBA $\mathcal{A}^- := (Q, \Sigma^-, q_0, \delta^-, \Omega)$ durch

$$\delta^-(q, a) \quad := \quad \{(q_1, \dots, \underbrace{q_i, \dots, q_i}_{m-i+1 \mathrm{mal}}) \mid \exists b \in \Sigma, \sigma(b) = i \text{ und } (q_1, \dots, q_i) \in \delta(q, b)\}$$

Beachte: $\mathcal{A}^-$ erkennt intuitiv all diejenigen Bäume, die $\mathcal{A}$ erkennen würde, ohne dabei auf das Alphabet zu achten.

Lemma 35
Sei $\mathcal{A}$ ein PBA über Σ. Es gilt $L(\mathcal{A}) = \emptyset$ gdw. $L(\mathcal{A}^-) = \emptyset$.

Beweis "$\Leftarrow$" Angenommen, $t \in L(\mathcal{A})$. Sei t' derjenige Baum, der aus t entsteht, indem jede Beschriftung durch a ersetzt wird, und der jeweils rechteste Sohn eines jeden Knoten so oft kopiert wird, bis jeder Knoten die Stelligkeit $\max\{\sigma(b) \mid b \in \Sigma\}$ hat. Man sieht leicht, dass $t' \in L(\mathcal{A}^-)$ gilt.

"⇒" Sei $m := \max\{\sigma(b) \mid b \in \Sigma\}$. Angenommen, $t' \in L(\mathcal{A}^-)$. Dann existiert ein erfolgreicher Lauf r von $\mathcal{A}^-$ auf t'. Sei $w \in dom(t')$ und $r(w) = q$ für ein $q \in Q$. Also existieren $q_1, \ldots, q_m$, so dass $r(w(i-1)) = q_i$ für $i = 1, \ldots, m$ gilt. Laut Definition von δ^- existiert dann auch ein $b \in \Sigma$, so dass $(q_1, \ldots, q_{\sigma(b)}) \in \delta(q, b)$ ist. In dem die $m - \sigma(b)$ rechten Söhne von jedem solchen w eliminiert werden, erhält man einen Baum t, auf dem r einen Lauf des Automaten $\mathcal{A}$ bildet. Da nur Pfade verworfen wurden, gilt die Paritätsbedingung weiterhin auf allen Pfaden, womit $t \in L(\mathcal{A})$ gezeigt ist. ■

Satz 66
Das Leerheitsproblem für PBAs mit n Zuständen und p Prioritäten über einem Alphabet Σ kann in Zeit $n^{O(p)}$ entschieden werden.

Beweis Sei $m := \max\{\sigma(b) \mid b \in \Sigma\}$ und $\mathcal{A}$ ein PBA mit n Zuständen und p Prioritäten. Offensichtlich ist dann auch $\mathcal{A}^-$ ein PBA mit denselben Größenbeschränkungen. Der Leerheitstest auf $\mathcal{A}$ reduziert sich laut Lemma 35 auf den Leerheitstest auf $\mathcal{A}^-$. Es gilt jedoch: $L(\mathcal{A}^-) \neq \emptyset$ gdw. $t_a \in L(\mathcal{A}^-)$, wobei t_a der Baum ist, dessen Knoten alle Beschriftung a und denselben Verzweigungsgrad m haben.

Laut Lemma 34 gilt $t_a \in L(\mathcal{A}^-)$ gdw. Spieler A das Paritätsspiel $G(\mathcal{A}^-, t_a)$ gewinnt. Da t_a fest ist, kann dieses Spiel als endlicher Graph mit $n+n^m$ Knoten und $O(n^{2m})$ vielen Kanten repräsentiert werden. Außerdem gibt es in $G(\mathcal{A}^-, t_a)$ nur p viele Prioritäten. Der in Abschnitt 5.3 skizzierte rekursive Algorithmus löst endliche Paritätsspiele in $O(e \cdot v^d)$ vielen Schritten, wobei e die Anzahl der Kanten, v die Anzahl der Knoten und d die maximale Priotität ist. Dies liefert einen Leerheitstest für PBA, der in Zeit $n^{O(p)}$ läuft, wenn die Stelligkeit des Alphabets fest ist. ■

Definition 47
Sei Σ ein Alphabet mit Stelligkeiten $\{1, \ldots, m\}$. Ein Baum über Σ heißt *endlich repräsentierbar*, falls er nur endlich viele nicht-isomorphe Teilbäume hat.

Beachte, dass die endlich repräsentierbaren ω-Bäume genau diejenigen sind, die sich durch Gleichungssysteme der Form

$$\begin{array}{rcl} t_1 & = & a_1(t_{i_{1,1}}, \ldots, t_{i_{1,\sigma(a_1)}}) \\ & \vdots & \\ t_n & = & a_n(t_{i_{n,1}}, \ldots, t_{i_{n,\sigma(a_n)}}) \end{array}$$

darstellen lassen.

Korollar 67
Sei $\mathcal{A}$ ein PBA. Es gilt $L(\mathcal{A}) \neq \emptyset$ gdw. es einen endlich repräsentierbaren Baum t gibt, so dass $t \in L(\mathcal{A})$ gilt.

Beweis Die Richtung "⇐" ist offensichtlich. Die Richtung "⇒" folgt aus Satz 66. Sei $\mathcal{A} = (Q, \Sigma, q_0, \delta, \Omega)$. Angenommen $L(\mathcal{A}) \neq \emptyset$. Dann hat Spieler A eine Gewinnstrategie s für das Spiel $G(\mathcal{A}^-, t_a)$. Da es sich um ein Paritätsspiel handelt, kann diese sogar als positional angenommen werden. Beachte: $s : Q \to Q^m$, wobei m die maximale Stelligkeit des Baumalphabets ist. Wir benutzen nun s, um einen endlich repräsentierbaren Baum zu konstruieren. Dieser ergibt sich als maximale Lösung für t_{q_0} im Gleichungssystem, welches für jedes $q \in Q$ die folgende Gleichung enthält.

$$t_q \;=\; b(t_{q_{i_1}}, \ldots, t_{q_{i_n}})$$

falls $\sigma(b) = n$ und $s(q) = (q_{i_1}, \ldots, q_{i_n}) \in \delta(q, b)$.

Zuerst zeigen wir, dass dieses Gleichungssystem existiert. Nach Voraussetzung ist $s(q_0)$ definiert. Also existiert ein $b \in \Sigma$ mit $s(q) = (q_{i_1}, \ldots, q_{i_n}) \in \delta(q_0, b)$. Da im Spiel $G(\mathcal{A}^-, t_a)$ nun Spieler P mit der Wahl eines q_{i_j} antwortet, muss $s(q_{i_j})$ wiederum für alle $j = 1, \ldots, n$ definiert sein. Dieser Prozess terminiert, da es nur endlich viele q gibt.

Es bleibt noch zu zeigen, dass $t_{q_0} \in L(\mathcal{A})$ gilt. Man sieht leicht, dass es einen Lauf von $\mathcal{A}$ auf t_{q_0} gibt – dieser beschriftet die Wurzel eines Unterbaums t_q mit dem Zustand q. Dass dieser Lauf erfolgreich ist, folgt aus der Tatsache, dass s eine Gewinnstrategie für Spieler A in $G(\mathcal{A}^-, t_a)$ ist. Beachte, dass jeder Pfad in diesem Lauf einer Partie in $G(\mathcal{A}^-, t_a)$ entspricht, in der Spieler A gemäß der Strategie s gewählt hat. Also ist die größte, unendlich auftretende Priorität in dieser Partie gerade. Die Prioritäten im Spiel sind jedoch die der Automatenzustände. Also ist auch die größte, unendlich oft auftretende Priorität auf jedem Pfad des Laufs gerade. ■

5.6 Unendliche Bäume als metrischer Raum

In diesem Abschnitt betrachten wir eine vielleicht nicht ganz offensichtliche, algebraische Eigenschaft der Menge der unendlichen, binären Bäume. Diese wird uns später helfen zu zeigen, dass PBAs mit mehr Prioritäten in gewissen Fällen auch mehr erkennen.

Definition 48
Ein *metrischer Raum* ist ein Paar (M, Δ), wobei M eine beliebige Menge und $\Delta : M \times M \to \mathbb{R}$ ist, so dass für alle $x, y, z \in M$ gilt:

1. $\Delta(x, y) \geq 0$, (Positivität)
2. $\Delta(x, y) = 0$ gdw. $x = y$, (Identität bei unsichtbarem Abstand)
3. $\Delta(x, y) = \Delta(y, x)$, (Symmetrie)
4. $\Delta(x, y) \leq \Delta(x, z) + \Delta(z, y)$. (Dreiecksungleichung)

Eine Folge $x_0, x_1, \ldots$ von Elementen in M heißt *Cauchy-Folge*, falls es für jedes $r \in \mathbb{R}$ mit $r > 0$ ein $n \in \mathbb{N}$ gibt, so dass $\Delta(x_i, x_j) < r$ für alle $i, j \geq n$.

Ein x heißt *Limes* einer Cauchy-Folge $x_0, x_1, \ldots$, falls es für alle $r \in \mathbb{R}$ mit $r > 0$ ein $n \in \mathbb{N}$ gibt, so dass $\Delta(x, x_i) < r$ für alle $i \geq n$. Bezeichnung: $x = \lim\limits_{i \to \infty} x_i$.

Ein metrischer Raum heißt *vollständig*, falls für jede Cauchy-Folge $(x)_i$ gilt: $\lim\limits_{i \to \infty} x_i \in M$.

Lemma 36
Sei $\mathcal{T}_\Sigma$ die Menge aller unendlichen, binären Bäume über Σ und $\Delta : \mathcal{T}_\Sigma^2 \to \mathbb{R}$ definiert wie folgt:

$$\Delta(t_1, t_2) \ := \ \inf \{2^{-k} \mid t_1(w) = t_2(w) \text{ für alle } w \in \{0,1\}^* \text{ mit } |w| \leq k\}$$

Dann ist $(\mathcal{T}_\Sigma, \Delta)$ ein vollständiger, metrischer Raum.

Beweis Übung. ■

Definition 49
Sei (M, Δ) ein metrischer Raum. Eine Abbildung $f : M \to M$ heißt *kontrahierend*, falls es ein $c \in \mathbb{R}$ gibt mit $c < 1$, so dass für alle $x, y \in M$ gilt: $\Delta(f(x), f(y)) \leq c \cdot \Delta(x, y)$.

Satz 68 (Banach)
Sei (M, Δ) ein vollständiger, metrischer Raum, $M \neq \emptyset$ und $f : M \to M$ eine kontrahierende Abbildung darauf. Dann hat f einen eindeutigen Fixpunkt.

Beweis Sei $c < 1$ die Kontraktionskonstante von f. Da $M \neq \emptyset$, existiert ein $x_0 \in M$. Definiere für alle $i \in \mathbb{N}$: $x_{i+1} := f(x_i)$ und $d_i := \Delta(x_i, x_{i+1})$. Nun gilt für alle $n, i \in \mathbb{N}$ mithilfe der Dreiecksungleichung:

$$\Delta(x_n, x_{n+i}) \ \leq \ \sum_{j=n}^{n+i-1} d_n$$

Man sieht leicht, dass die Reihe $\sum\limits_{j=0}^{\infty} d_j$ konvergiert.

$$\sum_{j=0}^{\infty} d_j \ \leq \ \sum_{j=0}^{\infty} c^j \cdot d_0 \ = \ d_0 \cdot \frac{1}{1-c}$$

da $c < 1$ gilt. Beachte, dass $d_{n+1} \leq c \cdot d_n$ für alle $n \in \mathbb{N}$ gilt. Somit ist $(x)_i$ eine Cauchy-Folge, denn angenommen, es gäbe ein $r \in \mathbb{R}$, so dass $r > 0$ und für alle $n \in \mathbb{N}$ es $i, j \geq n$ gibt, so dass $\Delta(x_i, x_j) \geq r$ wäre, dann betrachte die Teilfolge $(y)_i$ dieser unendlichen vielen x_i und x_j. Es gilt

$$\sum_{i=0}^{\infty} r \ \leq \ \sum_{i=0}^{\infty} \Delta(y_i, y_{i+1}) \ \leq \ \sum_{i=0}^{\infty} \Delta(x_i, x_{i+1})$$

mit der Dreiecksungleichung. Somit wäre obige Reihe nicht konvergent, denn $\sum_{i=0}^{\infty} r$ ist offensichtlich nicht konvergent.

Da M vollständig ist, existiert ein $x \in M$ mit $x = \lim\limits_{i \to \infty} x_i$. Wir behaupten, dass x Fixpunkt von f ist. Dazu zeigen wir zuerst, dass auch $f(x)$ Limes der Folge $(x)_i$ ist. Es gibt also für jedes $r \in \mathbb{R}$ ein $n \in \mathbb{N}$, so dass $\Delta(x, x_i) < r$ für alle $i \geq n$. Somit existiert für jedes $r \in \mathbb{R}$ ein $n \in \mathbb{N}$, so dass $\Delta(f(x), f(x_i)) = \Delta(f(x), x_{i+1}) < c \cdot r < r$ für alle $i \geq n$ gilt.

Als nächstes zeigen wir, dass der Abstand zwischen x und $f(x)$ nicht sichtbar ist. Beachte, dass wegen der Dreiecksungleichung für alle $i \in \mathbb{N}$ gilt:

$$\Delta(x, f(x)) \;\leq\; \Delta(x, x_i) + \Delta(x_i, f(x))$$

Also auch

$$\Delta(x, f(x)) \;\leq\; \lim_{i\to\infty} \big(\Delta(x, x_i) + \Delta(x_i, f(x))\big) \;=\; 0$$

wegen $\lim\limits_{i\to\infty} \Delta(x, x_i) = 0$ und $\lim\limits_{i\to\infty} \Delta(f(x), x_i) = 0$. Aus dem Axiom über die Identität bei unsichtbarem Abstand folgt dann $f(x) = x$, d.h. x ist wirklich Fixpunkt von f.

Letztlich bleibt noch zu zeigen, dass der Fixpunkt eindeutig ist. Angenommen, dies ist nicht der Fall. D.h. es gibt $x, y \in M$ mit $f(x) = x$ und $f(y) = y$. Dann gilt

$$\Delta(x, y) \;=\; \Delta(f(x), f(y)) \;\leq\; c \cdot \Delta(x, y)$$

Da $c < 1$, wird diese Ungleichung nur von $\Delta(x, y) = 0$ gelöst. Mit dem Axiom über die Identität bei unsichtbarem Abstand folgt dann aber $x = y$. ∎

5.7 MSO auf unendlichen Bäumen

In diesem Abschnitt betrachten wir Prädikatenlogik auf unendlichen Bäumen. Zur Erinnerung: Auf unendlichen Wörtern machte es keinen Unterschied, ob in der Signatur für MSO die totale Ordnung $\leq$ oder die Nachfolgerfunktion *succ* vorkommt, da MSO stark genug ist, das eine mittels des anderen auszudrücken. Andererseits ist FO[$\leq$] jedoch ausdruckstärker als FO[*succ*].

Auf Bäumen bildet $\leq$ keine totale Ordnung, sondern nur eine partielle, und eine einzige Nachfolgefunktion macht wenig Sinn. Aus diesem Grund beschränken wir uns auf Bäume mit einem festen, maximalen Verzweigungsgrad d und Nachfolgerfunktionen $succ_i$ für $i = 1, \dots, d$.

Zur Vereinfachung nehmen wir hier an, dass ω-Bäume immer binär und total sind. Dies ist keine Einschränkung, da sich jeder endlich verzweigende Baum als binärer Baum kodieren lässt.

Definition 50
Sei Σ ein Alphabet mit $\sigma(a) = 2$ für alle $a \in \Sigma$. Formeln der Logik *MSO auf binären Bäumen* oder auch *S2S* (Second-Order Logic of Two Successors) über zwei Mengen $\mathcal{V}_1 = \{x, y, \dots\}$ und $\mathcal{V}_2 = \{X, Y, \dots\}$ von erst- bzw. zweitstufigen Variablen sind gegeben durch die folgende Grammatik.

$$\varphi \;::=\; x = y \mid succ_i(x, y) \mid P_a(x) \mid X(x) \mid \varphi \vee \varphi \mid \neg\varphi \mid \exists x.\varphi \mid \exists X.\varphi$$

wobei $a \in \Sigma$, $x, y \in \mathcal{V}_1$, $X \in \mathcal{V}_2$ und $i \in \{1, 2\}$. Wie benutzen wieder die üblichen Abkürzungen.

Definition 51
Sei t ein totaler, binärer ω-Baum, d.h. $dom(t) = \{1,2\}^*$. Eine Umgebung ρ bildet erststufige Variablen x auf Knoten $\rho(x) \in \{1,2\}^*$ und zweitstufige Variablen X auf Mengen von Knoten $\rho(X) \subseteq \{1,2\}^*$ ab. Die Semantik von S2S ist wie folgt definiert.

$$
\begin{array}{rll}
t \models_\rho x = y & \text{gdw.} & \rho(x) = \rho(y) \\
t \models_\rho succ_i(x,y) & \text{gdw.} & \rho(y) = \rho(x)i \\
t \models_\rho P_a(x) & \text{gdw.} & t(\rho(x)) = a \\
t \models_\rho x = y & \text{gdw.} & \rho(x) = \rho(y) \\
t \models_\rho X(x) & \text{gdw.} & \rho(x) \in \rho(X) \\
t \models_\rho \varphi \vee \psi & \text{gdw.} & t \models_\rho \varphi \text{ oder } t \models_\rho \psi \\
t \models_\rho \neg\varphi & \text{gdw.} & t \not\models_\rho \varphi \\
t \models_\rho \exists x.\varphi & \text{gdw.} & \text{es gibt } w \in \{1,2\}^* \text{ so dass } t \models_{\rho[x \mapsto w]} \varphi \\
t \models_\rho \exists X.\varphi & \text{gdw.} & \text{es gibt } W \subseteq \{1,2\}^* \text{ so dass } t \models_{\rho[X \mapsto W]} \varphi
\end{array}
$$

Satz 69
Eine Baum-Sprache wird von einem PBA erkannt, gdw. sie S2S-definierbar ist.

Beweis Analog zum Satz von Büchi. Für die Richtung "⇒" beschreibt man in S2S wiederum die Existenz eines erfolgreiches Laufs eines PBA auf einem Baum. Beachte, dass sich die Aussage "auf allen Pfaden ist die maximale, unendlich oft auftretende Priorität gerade" in S2S leicht ausdrücken lässt, wenn die möglichen Prioritäten vorgegeben sind.

Für die Richtung "⇐" definiert man zuerst eine Normalform, in der keine erststufigen Variablen mehr vorkommen. Die Übersetzung einer solchen Formel $\varphi(X_1, \ldots, X_n)$ in einen PBA über dem Alphabet $\Sigma \times \{0,1\}^n$ erfolgt dann wieder induktiv. Vereinigung wird durch Nichtdeterminismus behandelt, Negation braucht den Komplementabschluss der von PBAs erkannten Sprachen (Satz 65). Existentielle Quantifizierung benutzt Alphabetprojektion, also den Abschluss unter Homomorphismen (Satz 61). ■

Korollar 70 (Rabin'69)
Die Logik S2S ist entscheidbar.

Beweis Folgt aus der Tatsache, dass die Übersetzungen in Satz 69 effektiv sind und das Leerheitsproblem für PBAs entscheidbar ist (Satz 66). ■

Rabins originaler Beweis ist etwas aufwändiger, da er den Komplementabschluss von Rabin-Baumautomaten gezeigt hat.

5.8 Der modale μ-Kalkül

Unschön an S2S ist die hohe Komplexität ihres Entscheidungsproblems. Daher betrachten wir eine Logik, deren Ausdrucksstärke fast dieselbe ist wie die von S2S, deren Komplexität jedoch geringer ist.

Definition 52
Sei V eine Menge zweitstufige Variablen. Formeln des *modalen μ-Kalkül* ($\mathcal{L}_\mu$) über unendlichen binaren Bäumen über einem Alphabet Σ sind durch folgende Grammatik gegeben.

$$\varphi \ ::= \ a \mid \varphi \vee \varphi \mid \neg\varphi \mid \Diamond\varphi \mid X \mid \mu X.\varphi$$

Neben den üblichen Abkürzungen benutzen wir auch $\Box\varphi := \neg\Diamond\neg\varphi$, $\nu X.\varphi := \neg\mu X.\neg\varphi[\neg X/X]$. Mit $Sub(\varphi)$ bezeichnen wir die Menge aller Unterformeln von φ. Wir definieren $|\varphi| := |Sub(\varphi)|$.

Seien $\mu X.\psi_1, \mu Y.\psi_2 \in Sub(\varphi)$. Wir schreiben $Y \prec_\varphi X$, falls X ungebunden in $Sub(\psi_2)$ vorkommt. Sei $<_\varphi$ die transitive Hülle von $\prec_\varphi$. Beachte, dass $<_\varphi$ irreflexiv ist bildet.

Eine Formel φ heißt *wohlgeformt*, wenn

1. jede Variable in φ höchstens einmal durch ein $\mu X.\varphi$ gebunden wird,
2. jedes Vorkommen einer Variablen X unter einer geraden Anzahl von Negationen innerhalb des entsprechenden $\mu X.\varphi$ liegt.

Wir gehen davon aus, dass Formeln immer wohlgeformt sind. Der Grund für diese Forderung ist der folgende: Der Operator μ quantifizert über Mengen von Knoten in dem Baum, allerdings nicht wie bei S2S existentiell oder universell. Beachte, dass eine Formel $\varphi(X)$ mit einer freien Variablen X eine Abbildung von Mengen von Knoten auf Mengen von Knoten darstellt. Ist φ wohlgeformt, dann ist diese Abbildung monoton und besitzt einen eindeutigen kleinsten Fixpunkt bzgl. der Ordung $\subseteq$. $\mu X.\varphi$ bezeichnet genau diesen. Genauso bezeichnet $\nu X.\varphi$ den größten Fixpunkt dieser Abbildung. Dies wird in der folgenden Definition formalisiert.

Definition 53
Sei t ein totaler, binärer Baum über Σ. Die Semantik einer $\mathcal{L}_\mu$-Formel φ ist induktiv definiert wie folgt. Wir benutzten wieder eine Umgebung $\rho : V \to 2^{\{0,1\}^*}$, um freie Variablen zu interpretieren.

$$\begin{aligned}
[\![a]\!]^t_\rho &:= \{w \in \{0,1\}^* \mid t(w) = a\}\\
[\![\varphi \vee \psi]\!]^t_\rho &:= [\![\varphi \vee \psi]\!]^t_\rho \cup [\![\varphi \vee \psi]\!]^t_\rho\\
[\![\neg\varphi]\!]^t_\rho &:= \{0,1\}^* \setminus [\![\varphi]\!]^t_\rho\\
[\![\Diamond\varphi]\!]^t_\rho &:= \{w \in \{0,1\}^* \mid \exists x \in \{0,1\}, wx \in [\![\varphi]\!]^t_\rho\}\\
[\![X]\!]^t_\rho &:= \rho(X)\\
[\![\mu X.\varphi]\!]^t_\rho &:= \bigcap\{T \subseteq \{0,1\}^* \mid [\![\varphi]\!]^t_{\rho[X\mapsto T]} \subseteq T\}
\end{aligned}$$

Die letzte Klausel benutzt dabei den Satz von Knaster-Tarski, welcher besagt, dass der kleinste Fixpunkt einer montonen Abbildung immer existiert und dem Infimum aller Präfixpunkte gleicht.

Wir schreiben $t \in L_\rho(\varphi)$, falls $\epsilon \in [\![\varphi]\!]^\rho_t$ und lassen ρ auch weg, wenn φ geschlossen ist. Zwei Formeln sind äquivalent, $\varphi \equiv \psi$, falls für alle t und alle ρ gilt: $[\![\varphi]\!]^t_\rho = [\![\psi]\!]^t_\rho$.

Beispiel 14
Die Fixpunktoperatoren liest man am einfachsten als Rekursion. Dabie bedeutet μ in etwa, dass die Rekursion nach endlichen vielen Schritten terminieren muss, während eine Rekursion durch eine ν-gebundene Variable ins Unendliche laufen darf. Außerdem sind Variablen, die weiter außen stehen, wichtiger als innere. So darf also z.B. eine ν-Rekursion unendlich oft eine μ-Rekursion starten, aber nicht umgekehrt.

- $\mu X.a \vee \Diamond X$ besagt "irgendwo kommt ein a vor".
- $\mu X.a \vee \Box X$ besagt "auf allen Pfaden kommt irgendwann ein a vor".
- $\nu X.a \wedge \Box X$ besagt "der gesamte Baum ist überall mit a beschriftet".
- $\nu X.\mu Y.(a \wedge \psi \wedge \Diamond X) \vee (\psi \wedge \Diamond Y)$ besagt "es gibt einen Pfad, auf dem überall ψ erfüllt ist und a unendlich oft vorkommt".
- $\nu X.(\psi \wedge \Box X) \vee (\mu Y.\nu Z.\Box Y \vee (\neg a \wedge \Box Z))$ besagt "auf allen Pfaden, auf denen unendlich oft a vorkommt, gilt überall ψ".

In den letzten beiden Beispielen muss ψ jeweils eine geschlossene Formel sein.

Beispiel 15
Betrachte die folgenden zwei Formeln: $\varphi_1 := \nu X.(\mu Y.a \vee \Box Y) \wedge \Box X$ und $\varphi_2 := \nu X.\mu Y.(a \vee \Box Y) \wedge \Box X$. Komischerweise sagen beide dasselbe aus: "auf allen Pfaden kommen unendlich viele a vor". Dennoch ist φ_1 in gewissem Sinne die "bessere" Formel. Die innere Fixpunktformel darin ist geschlossen, d.h. ihre Semantik in einem Modell hängt nicht von einer Belegung für X ab. So kann man diese zuerst mithilfe einer Fixpunktiteration berechnen, um dann die äußere Fixpunktformel ebenfalls mit einer Fixpunktiteration zu berechnen. Angenommen, ein endlich-repräsentierten Baum mit n Unterbäumen liegt vor. Dann kann man die Semantik von φ_1 in $O(n)$ Iteration berechnen.

In φ_2 hängt die Semantik der inneren Formel jedoch von X ab. Berechnet man nun die Semantik von φ_2, dann braucht man auf solchen Modellen $O(n)$ Iterationsschritte für die äußere Fixpunktformel. In jedem Schritt ändert sich jedoch X, so dass man jedes Mal die Semantik der inneren Fixpunktformel neu berechnen muss, was ebenfalls $O(n)$ viele Schritte benötigt. Dadurch braucht man insgesamt $O(n^2)$ viele Schritte.

Der Grund dafür ist also die Verschachtelung von Fixpunktformeln. Sind diese vom selben Typ, dann kann man ein optimiertes Verfahren anwenden, welches ebenfalls nur $O(n)$ Schritte in diesem Beispiel brauchen würde. Sind diese aber von verschiedenem Typ, dann ist dies nicht möglich. Die Verschachtelung verschiedenartiger Fixpunktformeln bezeichnet man als *Alternierung*.

Definition 54
Eine $\mathcal{L}_\mu$-Formel ist in *positiver Normalform*, wenn sie nur aus Literalen a und $\neg a$, Variablen X und den Konstruktoren $\vee, \wedge, \Diamond, \Box, \mu$ und ν aufgebaut ist.

Lemma 37
Jede wohlgeformte $\mathcal{L}_\mu$-Formel φ ist äquivalent zu einem φ' in positiver Normalform, so dass φ' effektiv aus φ konstruiert werden kann und $|\varphi'| = O(|\varphi|)$ gilt.

Mit $\overline{\varphi}$ bezeichnen wir die eindeutige positive Normalform der Formel $\neg\varphi$.

5.8.1 Der μ-Kalkül und Paritätsspiele

Definition 55
Sei t ein unendlicher, binärer Baum und φ_0 eine wohl-geformte $\mathcal{L}_\mu$-Formel in positiver Normalform. Das *Model-Checking-Spiel* $G(\varphi, t)$ wird zwischen Spielern A und P auf Positionen $(w, \psi) \in \{0,1\}^* \times \mathit{Sub}(\varphi_0)$ wie folgt gespielt. Jede Partie beginnt in der Position (ϵ, φ_0). Ist eine partielle Partie $(w_0, \varphi_0), \ldots, (w_i, \varphi_i)$ bereits konstruiert worden, dann bestimmt φ_i den nächsten Zug.

- Ist $\varphi_i = a$, so ist die Partie beendet.
- Ist $\varphi_i = \neg a$, so ist die Partie beendet.
- Ist $\varphi_i = \psi_1 \vee \psi_2$, dann wählt Spieler A ein $j \in \{1, 2\}$ und $(w_{i+1}, \varphi_{i+1}) := (w_i, \psi_j)$.
- Ist $\varphi_i = \psi_1 \wedge \psi_2$, dann wählt Spieler P ein $j \in \{1, 2\}$ und $(w_{i+1}, \varphi_{i+1}) := (w_i, \psi_j)$.
- Ist $\varphi_i = \Diamond\psi$, dann wählt Spieler A ein $x \in \{0, 1\}$ und $(w_{i+1}, \varphi_{i+1}) := (w_i x, \psi)$.
- Ist $\varphi_i = \Box\psi$, dann wählt Spieler P ein $x \in \{0, 1\}$ und $(w_{i+1}, \varphi_{i+1}) := (w_i x, \psi)$.
- Ist $\varphi_i = X$, und wird X gebunden durch $\sigma X.\psi$ in φ_0, dann ist $(w_{i+1}, \varphi_{i+1}) := (w_i, \psi)$.
- Ist $\varphi_i = \sigma X.\psi$, dann ist $(w_{i+1}, \varphi_{i+1}) := (w_i, X)$.

Spieler A gewinnt endliche Partien, die in einer Position (w, a) mit $a = t(w)$ oder einer Position $(w, \neg a)$ mit $a \neq t(w)$ enden. Spieler P gewinnt eine endliche Partie in allen anderen Fällen.

Beachte, dass es in jeder unendlichen Partie mindestens eine Variable geben muss, die unendlich oft auftritt. Außerdem gilt: wenn zwei Variablen X und Y unendlich oft auftreten, dann muss auch entweder $X <_\varphi Y$ oder $Y <_\varphi X$ gelten. Da $<_\varphi$ irreflexiv ist, kann nicht beides gelten. Da $<_\varphi$ transitiv ist, gibt es in jeder unendlichen Partie eine größte (bzgl. $<_\varphi$) Variable, die unendlich oft auftritt. Spieler A gewinnt solch eine Partie, wenn diese eindeutige Variable durch ein ν gebunden ist. Ist sie durch ein μ gebunden, dann gewinnt Spieler P.

Man kann zeigen, dass diese Spiele korrekt die Frage, ob ein gegebener Baum in der Sprache einer gegebenen Formel ist, charakterisieren. Dazu braucht man jedoch ein wenig Fixpunkttheorie, weswegen wir hier auf den Beweis verzichten wollen.

Satz 71
Für alle Bäume t und alle geschlossenen $\mathcal{L}_\mu$-Formeln φ gilt: Spieler A hat eine Gewinnstrategie für das Spiel $G(\varphi, t)$ gdw. $t \in L(\varphi)$.

Viel wichtiger ist die Beobachtung, dass es sich bei $G(\varphi, t)$ im Grunde um ein Paritätsspiel handelt. Die Partitionierung in Positionen erhält man, indem die Positionen, in denen laut Regeln deterministisch weitergespielt wird, beliebig einem Spieler zuschlägt. Positionen, in denen eine Partie endet, kann man in sich selbst ins Unendliche fortführen.

Gewinnt dort Spieler A, so erhält diese Position die Priorität 0, anderernfalls die Priorität 1. Positionen der Form (w, X) erhalten eine gerade Priorität, falls X durch ν gebunden wird, ansonsten eine ungerade. Außerdem muss diese verträglich mit der Ordnung $<_\varphi$ sein. D.h. wenn $X <_\varphi Y$ gilt, dann muss die Priorität von einem (w, Y) mindestens so groß sein wie die von einem (v, X).

5.8.2 Die Ausdrucksstärke des modalen μ-Kalküls

Satz 72
$\mathcal{L}_\mu \leq$ *S2S.*

Beweis Dies ist auf den ersten Blick evtl. nicht offensichtlich, aber dennoch erstaunlich einfach zu zeigen. Man kann zu jeder $\mathcal{L}_\mu$-Formel φ induktiv eine S2S-Formel $tr(\varphi)_x$ mit einer freien erststufigen Variablen x angeben, so dass für alle Bäume t, alle $w \in \{0,1\}^*$ und alle ρ gilt:

$$w \in [\![\varphi]\!]_t^\rho \quad \text{gdw.} \quad t, w \models_\rho tr(\varphi)_x$$

Dabei interpretiert ρ die freien und gleichnamigen zweitstufigen Variablen in $tr(\varphi)_x$ genauso wie in φ, und x wird durch w interpretiert.

$$\begin{aligned}
tr_x(a) &:= P_a(x) \\
tr_x(\varphi \vee \psi) &:= tr_x(\varphi) \vee tr_x(\psi) \\
tr_x(\neg\varphi) &:= \neg tr_x(\varphi) \\
tr_x(\Diamond\varphi) &:= \exists y.(succ_0(x,y) \vee succ_1(x,y)) \wedge tr_y(\varphi) \\
tr_x(X) &:= X(x) \\
tr_x(\mu X.\varphi) &:= \forall X.\big(\forall y.tr_y(\varphi) \to X(y)\big) \to X(x)
\end{aligned}$$

Die Korrektheit dieser Übersetzung folgt sofort aus der Tatsache, dass diese lediglich die Semantik von $\mathcal{L}_\mu$ in S2S formalisiert.

Es stellt sich die Frage, ob auch die Umkehrung gilt. Zwar lässt sich jede MSO-Formel auf Wörtern in eine $\mathcal{L}_\mu$-Formel auf Wörtern übersetzen. Auf Bäumen ist dies jedoch nicht möglich.

Definition 56
Seien t, t' zwei Bäume über einem Alphabet Σ. Wir schreiben $t \sim t'$, falls es eine Bijektion $\gamma : \{0,1\}^* \to \{0,1\}^*$ gibt mit folgenden Eigenschaften:

- Für alle $w \in \{0,1\}^*$ ist $t(w) = t'(\gamma(w))$.
- Für alle $w \in \{0,1\}^*$ existiert ein $x \in \{0,1\}$, so dass gilt: $\gamma(w0) = \gamma(w)x$ und $\gamma(w1) = \gamma(w)(1-x)$.

Es gilt also $t \sim t'$, wenn sich t und t' nur durch die Ordnung ihrer Unterbäume unterscheiden.

Eine Baumsprache L heißt *bisimulations-invariant*[1], falls für alle t, t' gilt: wenn $t \sim t'$, dann $t \in L$ gdw. $t' \in L$.

Lemma 38
Für alle geschlossenen $\mathcal{L}_\mu$-Formeln φ gilt: $L(\varphi)$ ist bisimulations-invariant.

Beweis Per Induktion über den Formelaufbau. Übung. ■

Daraus folgt sofort, dass S2S ausdrucksstärker ist als $\mathcal{L}_\mu$. So ist z.B. die Sprache $\{t \mid$ der linkeste Ast in t besteht nur aus a's $\}$ offensichtlich nicht bisimulations-invariant, aber S2S-definierbar, da man leicht einen PBA dafür angeben kann.

Korollar 73
$\mathcal{L}_\mu \lneq$ *S2S.*

Andererseits verliert $\mathcal{L}_\mu$ nicht zuviel Ausdruckstärke gegenüber S2S.

Satz 74
Eine Baumsprache ist $\mathcal{L}_\mu$-definierbar, gdw. sie S2S-definierbar und bisimulationsinvariant ist.

Beweis Es bleibt lediglich noch die Richtung $\Leftarrow$ zu zeigen. Angenommen L ist S2S-definierbar. Also existiert eine S2S-Formel φ mit $L(\varphi) = L$. Laut Satz 69 existiert dann auch ein PBA $\mathcal{A}_\varphi$ mit $L(\mathcal{A}_\varphi) = L$.

Sei $\mathcal{A}_\varphi = (Q, \Sigma, q_0, \delta, \Omega)$. Wir benutzen nun die Bisimulationsinvarianz von L, um $\mathcal{A}_\varphi$ in eine symmetrische Form zu bringen. Definiere $\mathcal{A}'_\varphi := (Q, \Sigma, q_0, \delta', \Omega)$ mit

$$\delta'_a(q) \;:=\; \{(q_1, q_2) \mid (q_1, q_2) \in \delta_a(q) \text{ oder } (q_2, q_1) \in \delta_a(q)\}$$

Es sollte klar sein, dass $L(\mathcal{A}'_\varphi) = \{t \mid \exists t' \in L(\mathcal{A}_\varphi), t \sim t'\}$. Da L aber bisimulations-invariant nach Voraussetzung ist, ist dies wiederum L selbst. Wir schreiben zur Vereinfachung $\{q_1; q_2\}$ für ein ungeordnetes Paar. Beachte, dass dies keine Menge ist, denn es hat immer zwei Elemente, auch wenn diese gleich sein können.

Es bleibt zu zeigen, dass sich $\mathcal{A}'_\varphi$ in $\mathcal{L}_\mu$ übersetzen lässt. O.B.d.A. nehmen wir an, dass $Q = \{q_0, \dots, q_n\}$ so geordnet ist, dass $\Omega(q_i) \geq \Omega(q_{i-1})$ für alle $i = 1, \dots, n$ gilt. Wir definieren nun für $i = n, \dots, 0$ jeweils eine Formel $\psi_i(X_0, \dots, X_{i-1})$ die intuitiv in allen Knoten eines Baumes gilt, die in einem erfolgreichen Lauf mit q_i beschriftet sind, solange die freien Variablen X_j angeben, welche Knoten mit Zuständen höherer Priorität beschriftet sind.

$$\psi_i(X_0, \dots, X_{i-1}) \;:=\; \sigma_i X_i. \bigwedge_{a \in \Sigma} a \to \bigvee_{\{q_j; q_h\} \in \delta_a(q_i)} \Diamond \alpha_{i,j} \wedge \Diamond \alpha_{i,h} \wedge \Box(\alpha_{i,j} \vee \alpha_{i,h})$$

wobei $\sigma_i = \mu$, falls $\Omega(q_i)$ ungerade und $\sigma_i = \nu$ sonst, und

$$\alpha_{i,j} \;:=\; \begin{cases} \psi_j & \text{, falls } j > i \\ X_j & \text{, falls } j \leq i \end{cases}$$

[1] Bisimulation wird eigentlich auf beliebigen, gerichteten Graphen definiert. Auf binären Bäumen ist die übliche Definition jedoch äquivalent zu der hier präsentierten.

Es sollte klar sein, dass ψ_0 eine geschlossene Formel ist. Man kann jetzt zeigen, dass $L(\psi_0) = L(\mathcal{A}'_\varphi)$ gilt, indem man für einen Baum t die beiden Spiele $G(\mathcal{A}'_\varphi, t)$ (entscheide, ob t von $\mathcal{A}'_\varphi$ erkannt wird) und das Spiel $G(\psi_0, t)$ (entscheide, ob $t \in L(\psi_0)$ ist) betrachtet. Beides sind Paritätsspiele von ähnlicher Struktur, und es gilt, dass Spieler A das eine gewinnt gdw. er das andere gewinnt. ∎

5.8.3 Ein Hierarchieresultat

Im folgenden widmen wir uns der Frage, ob sich jede Formel so umschreiben lässt, dass Alternierung entfernt wird. Dazu müssen wir zuerst Alternierung formal fassen. Wir gehen davon aus, dass Formeln immer in positiver Normalform vorliegen.

Definition 57
Die *Niwiński-Hierarchie* ist wie folgt aufgebaut.

- $\Sigma_0^{syn} = \Pi_0^{syn}$ sind alle Formeln, in denen keine Fixpunktoperatoren vorkommen.
- Σ_{n+1}^{syn} ist die kleinste Menge von Formeln, die $\Sigma_n^{syn} \cup \Pi_n^{syn}$ enthält, und für die gilt:
  - Ist $\varphi \in \Sigma_{n+1}^{syn}$, dann ist auch $\mu X.\varphi \in \Sigma_{n+1}^{syn}$ für jedes X.
  - Sind $\varphi(X_1, \ldots, X_k) \in \Sigma_{n+1}^{syn}$ und $\psi_i \in \Sigma_{n+1}^{syn}$ für $i = 1, \ldots, k$, dann ist auch $\varphi(\psi_1, \ldots, \psi_k) \in \Sigma_{n+1}^{syn}$, solange keine freie Variable von einem ψ_i durch das Einsetzen in φ gebunden wird.
- Π_{n+1}^{syn} ist so wie Σ_{n+1}^{syn} definiert mit ν statt mit μ.

Daraus leiten wir nun in einfacher Weise eine semantische Hierarchie ab.

Definition 58
Für alle $n \in \mathbb{N}$ sei

$$\begin{aligned} \Sigma_n &:= \{\varphi \mid \exists \psi \in \Sigma_n^{syn} \text{ mit } \varphi \equiv \psi\} \\ \Pi_n &:= \{\varphi \mid \exists \psi \in \Pi_n^{syn} \text{ mit } \varphi \equiv \psi\} \end{aligned}$$

Offensichtlich gilt $\Sigma_0 \subseteq \Sigma_1 \subseteq \Sigma_2 \subseteq \ldots$. Die Frage, der wir nachgehen, ist also, ob – und wenn ja, welche – diese Inklusionen strikt sind. Dass die erste strikt ist, ist leicht zu sehen. Ohne Fixpunktoperaroten kann man nur endlich tief in einen Baum hineinsehen und deswegen z.B. sicherlich nicht ausdrücken, dass irgendwo ein a vorkommt.

Lemma 39
Für alle $n \in \mathbb{N}$ und alle geschlossenen $\varphi \in \mathcal{L}_\mu$ gilt: $\varphi \in \Sigma_n^{syn}$ gdw. $\overline{\varphi} \in \Pi_n^{syn}$.

Beweis Dies folgt sofort daraus, dass das Übersetzen von $\neg\varphi$ in positive Normalform die Fixpunktoperatoren vertauscht. ∎

Jetzt betrachten wir noch einmal die Model Checking Spiele für $\mathcal{L}_\mu$. Wie oben angedeutet, kann man davon ausgehen, dass das Spiel $G(\varphi, t)$ als Paritätsspiel $(Pos_A, Pos_P, \delta, \Omega)$ gegeben ist. Beachte, dass jedes solche Spiel wiederum als binärer, unendlicher Baum aufgefasst über dem Alphabet $S_p := \{A, P\} \times \{0, \dots, p\}$ aufgefasst werden kann, wobei p die maximale vorkommende Priorität ist. Definiere

$$G(\varphi, t)(v) := (X, i) \quad \text{gdw.} \quad v \in Pos_X \text{ und } \Omega(v) = i$$

Es gilt sogar noch die folgende, etwas stärkere Aussage.

Lemma 40
Für alle Bäume t über einem beliebigen Alphabet, alle $n \in \mathbb{N}$ und alle geschlossenen $\varphi \in \Sigma_n^{syn}$ gilt: $G(\varphi, t)$ ist ein Baum über dem Alphabet S_n.

Definition 59
Für alle $n \in \mathbb{N}$ definiere die n-te *Walukiewicz-Formel* als

$$\Phi_n \;:=\; \sigma_n X_n . \sigma_{n-1} X_{n-1} . \dots \nu X_0 . \bigwedge_{i=0}^{n} \big((A, i) \to \Diamond X_i\big) \land \big((P, i) \to \Box X_i\big)$$

wobei für alle $i = 0, \dots, n$: $\sigma_i = \nu$, falls i gerade ist, und $\sigma_i = \mu$ sonst.

Lemma 41
Für alle $n \in \mathbb{N}$ gilt: falls n gerade ist, dann ist $\Phi_n \in \Pi_{n+1}$, und falls n ungerade ist, dann ist $\Phi_n \in \Sigma_{n+1}$.

Beweis Offensichtlich, da Φ_n bereits in der syntaktischen Hierarchie im entsprechenen Σ_{n+1}^{syn} bzw. Π_{n+1}^{syn} liegt. ■

Oben wurde gezeigt, dass das Problem zu entscheiden, ob ein Baum eine $\mathcal{L}_\mu$-Formel erfüllt, auf das Paritätsspielproblem reduziert werden kann. Die Walukiewicz-Formeln liefern die Umkehrung. Beachte, dass diese so konstruiert sind, dass sie die Existenz einer Gewinnstrategie für den Spieler A beschreiben.

Satz 75
Für alle Bäume t, alle $n \in \mathbb{N}$ und alle geschlossenen $\varphi \in \Sigma_n^{syn}$ gilt: $t \models \varphi$ gdw. $G(\varphi, t) \models \Phi_{n+1}$.

Beweis Übung. ■

Lemma 42
Für alle $n \in \mathbb{N}$ und alle geschlossenen $\varphi \in \Sigma_n$ ist die Abbildung $\lambda t.G(\varphi, t)$, die jeden Baum auf den Paritätsspielbaum auf φ abbildet, kontrahierend.

Beweis Übung. ■

Satz 76
Für alle $n \geq 1$ gilt: $\Sigma_{n-1} \subsetneq \Sigma_n$.

Beweis Die Inklusion ist klar. Es bleibt zu zeigen, dass diese auch strikt ist. Wir betrachten zuerst den Fall, dass n ungerade ist.

Nach Lemma 41 gilt $\Phi_{n+1} \in \Sigma_n$. Angenommen, es würde auch $\Phi_{n+1} \in \Sigma_{n-1}$ gelten. Per Definition gibt es dann also ein $\psi \in \Sigma^{syn}_{n-1}$, so dass $\psi \equiv \Phi_{n+1}$ ist. Nach Lemma 39 ist also $\overline{\psi} \in \Pi^{syn}_{n-1}$. Da aber $\Pi^{syn}_{n-1} \subseteq \Sigma^{syn}_n$ ist, gilt also ebenfalls $\overline{\psi} \in \Sigma^{syn}_n$.

Betrachte nun die Abbildung $\lambda t.G(\overline{\psi}, t)$. Laut Lemma 42 ist diese kontrahierend, und nach Banach's Fixpunktsatz hat sie also einen eindeutigen Fixpunkt t^*. Jetzt gilt folgendes:

$$t^* \models \Phi_{n+1} \text{ gdw. } t^* \models \psi \text{ gdw. } t^* \not\models \overline{\psi} \text{ gdw. } G(\overline{\psi}, t^*) \not\models \Phi_{n+1} \text{ gdw. } t^* \not\models \Phi_{n+1}$$

laut Satz 75, was offensichtlich ein Widerspruch ist.

Der Fall von n gerade wird genauso gezeigt mit einer entsprechenden Formulierung von Satz 75. Dabei vertauschen sich lediglich die Σ und Π. ∎

Diese Hierarchie überträgt sich auf die Baumautomaten. Mit n-PBA bezeichnen wir einen PBA, dessen maximale Priorität n ist.

Korollar 77
Für alle $n \in \mathbb{N}$ existiert eine reguläre Baumsprache L_n, die nicht von einem n-PBA erkannt wird.

Beweis Beachte, dass die Walukiewicz-Formeln Φ_n erfüllbar sind. Sei n ungerade. Definiere $L_n := L(\Phi_{n+2})$. Laut Sätzen 72 und 69 wird L_n von einem PBA $\mathcal{A}$ erkannt. Laut Lemma 38 ist L_n bisimulations-invariant. Nun sei angenommen, dass $\mathcal{A}$ ein n-PBA ist. Laut Satz 74 existiert dann auch eine $\mathcal{L}_\mu$-Formel ψ, so dass $L_n = L(\psi)$ ist. Damit wäre $\psi \equiv \Phi_{n+2}$. Der Beweis von Satz 74 zeigt jedoch auch, dass $\psi \in \Sigma^{syn}_n$ angenommen werden kann. Dies ist aber ein Widerspruch zu Satz 76.

Der Fall mit geradem n wird wiederum genauso mit Π statt Σ behandelt. ∎

Daraus folgt also insbesondere, dass Büchi-Baumautomaten weniger erkennen als Paritätsbaumautomaten. Beachte, dass sich die Hierarchie nicht gänzlich übertragen lässt. Man kann so nur zeigen, dass es Sprachen gibt, die nicht von einem n-PBA erkannt werden. Zwar werden diese von einem PBA erkannt, aber nicht notwendigerweise von einem $(n+1)$-PBA. Der Grund dafür ist die Tatsache, dass die Komplementierung von PBAs nicht die Anzahl der Prioritäten erhält.

Wir bemerken, dass man alternierende Baumautomaten definieren kann, die sich leichter komplementieren lassen, wobei die Anzahl der Prioritäten gleich bleibt. Für diese Automaten gilt dann ebenfalls wieder, dass die Hierarchie der mit n Prioritäten erkennbaren Sprachen überall strikt ist. Dies ist auch nicht verwunderlich, denn $\mathcal{L}_\mu$-Formeln sind im Grunde spezielle, alternierende PBAs.