Spiele für den Linearzeit μ-Kalkül

Christian Dax

Betreuer: Dr. Martin Lange

Kurzer Rückblick

Linearzeit μ-Kalkül

Definition (μTL)

$$\varphi ::= a \mid X \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid O\varphi \mid \mu X.\varphi \mid \nu X.\varphi$$

Example

- $\nu X.a \wedge OOX$ (an jd. 2. Stelle gilt a)
- $\mu X.a \vee OX$ (finally a)

Wort-Spiel mit Bsp.

Definition (Wort-Spiel)

- Start: $w \vdash \varphi$
- Regeln:

$$(\vee)\ \frac{w \vdash \varphi_0 \vee \varphi_1}{w \vdash \varphi_k}\quad \exists \qquad\qquad (\wedge)\ \frac{w \vdash \varphi_0 \wedge \varphi_1}{w \vdash \varphi_k}\quad \forall$$

$$(\mu)\ \frac{w \vdash \mu X.\varphi}{w \vdash X} \qquad\qquad (\nu)\ \frac{w \vdash \nu X.\varphi}{w \vdash X}$$

$$(O)\ \frac{w \vdash O\varphi}{w^1 \vdash \varphi} \qquad\qquad (X)\ \frac{w \vdash X}{w \vdash fb(\varphi)}$$

Wort-Spiel mit Bsp.

Definition

- Wer gewinnt die Partie?

  Spieler $\exists$ gewinnt:

  1. Partie endet mit $aw \vdash a$
  2. ∞-Partie: äußerste der ∞-Vars ist νX.

  Spieler $\forall$ gewinnt:

  3. Partie endet mit $aw \vdash b$
  4. ∞-Partie: äußerste der ∞-Vars ist μX.

Lemma

$\exists$ *gewinnt* *gdw.* $w \models \varphi$

Example

$$
\begin{array}{ll}
(ab)^{\omega} \vdash \mu Y.\nu X.(a \vee OOX) \wedge (b \vee OY) & \\ \hline
abab\cdots \vdash Y & \\ \hline
abab\cdots \vdash \nu X.(a \vee OOX) \wedge (b \vee OY) & \\ \hline
abab\cdots \vdash X & \\ \hline
abab\cdots \vdash (a \vee OOX) \wedge (b \vee OY) & \\ \hline
abab\cdots \vdash (b \vee OY) & \forall \\ \hline
abab\cdots \vdash \mathbf{O}Y & \exists \\ \hline
bab\cdots \vdash Y & \\ \hline
bab\cdots \vdash \nu X.(a \vee OOX) \wedge (b \vee OY) & \\ \hline
bab\cdots \vdash X & \\ \hline
bab\cdots \vdash (a \vee OOX) \wedge (b \vee OY) & \\ \hline
bab\cdots \vdash (a \vee OOX) & \forall \\ \hline
bab\cdots \vdash \mathbf{O}OX & \exists \\ \hline
ab\cdots \vdash \mathbf{O}X & \\ \hline
b\cdots \vdash X &
\end{array}
$$

Baum-Spiel

Definition (Baumspiel)

- Start: $t \vdash \varphi$
- Regeln:

$$(\vee)\ \frac{t \vdash \varphi_0 \vee \varphi_1, \Phi}{t \vdash \varphi_0, \varphi_1, \Phi} \qquad (\wedge)\ \frac{t \vdash \varphi_0 \wedge \varphi_1, \Phi}{t \vdash \varphi_k, \Phi}\ \ \forall$$

$$(\mu/\nu)\ \frac{t \vdash \mu/\nu X.\varphi, \Phi}{t \vdash X, \Phi} \qquad (X)\ \frac{t \vdash X, \Phi}{t \vdash fb(\varphi), \Phi}$$

$$(O)\ \frac{t \vdash O\varphi_0, \ldots, O\varphi_m, a_0, \ldots a_n}{t^1, \quad \varphi_0, \ldots, \varphi_m}\quad \forall : t^1$$

◂ Val-Spiel

Baum-Spiel

Definition

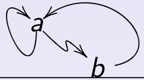
- Spieler $\exists$ gewinnt:
  1. Parie endet mit $\overset{a}{\wedge} \vdash a, \Phi$
  2. Partie hat ν-Linie
- Spieler $\forall$ gewinnt:
  3. Partie endet mit $\overset{a}{\wedge} \vdash b_1, \ldots, b_n$
  4. Partie hat keine ν-Linie

Lemma

$\exists$ *gewinnt* $\quad \Leftrightarrow \quad t \models \varphi$

Val-Spiel

Example



$$\vdash \quad \mu Y.\nu X.(a \vee OOX) \wedge (b \vee OY)$$

$$\overset{a}{\wedge} \vdash Y$$

$$\overset{a}{\wedge} \vdash \nu X.(a \vee OOX) \wedge (b \vee OY)$$

$$\overset{a}{\wedge} \vdash X$$

$$\overset{a}{\wedge} \vdash (a \vee OOX) \wedge (b \vee OY) \quad \forall$$

$$\overset{a}{\wedge} \vdash (a \vee OOX) \qquad \overset{a}{\wedge} \vdash (b \vee OY)$$

$$\overset{a}{\wedge} \vdash a, OOX \qquad \overset{a}{\wedge} \vdash b, \mathbf{O}Y$$

$$\forall : a$$

$$\overset{a}{\wedge} \vdash Y$$

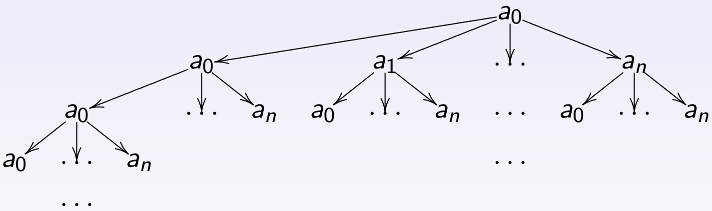
◂ ν-Line

Spiele für Allgemeingültigkeit und Erfüllbarkeit

Sigma Baum

Definition (Sigma Baum)

- $\Sigma = \{a_0, a_1, \ldots, a_n\}$
- $t_\Sigma :=$



Spiel auf Sigma Baum

Lemma

$$\exists \textit{ gewinnt } MC(t_\Sigma, O\varphi)$$

$$\begin{aligned}
&\Leftrightarrow \quad \textit{für alle Pfade } w \textit{ in } t_\Sigma : w \models O\varphi \\
&\Leftrightarrow \quad \textit{für alle Pfade } w^1 \textit{ (ohne Kopf) in } t_\Sigma : w^1 \models \varphi \\
&\Leftrightarrow \quad \textit{für alle Pfade } v \textit{ in } \Sigma^\omega : v \models \varphi
\end{aligned}$$

$$\Leftrightarrow \quad \varphi \textit{ allgemeingültig}$$

VAL-Spiel

Definition (Val-Spiel)

- Start: φ
- Regeln:

$$(\vee)\ \frac{\varphi_0 \vee \varphi_1, \Phi}{\varphi_0, \varphi_1, \Phi} \qquad\qquad (\wedge)\ \frac{\varphi_0 \wedge \varphi_1, \Phi}{\varphi_k, \Phi} \quad \forall$$

$$(\mu/\nu)\ \frac{\mu/\nu X.\varphi, \Phi}{X, \Phi} \qquad\qquad (X)\ \frac{X, \Phi}{fb(\varphi), \Phi}$$

$$(O)\ \frac{O\varphi_0, \ldots, O\varphi_m, a_0, \ldots a_n}{\varphi_0, \ldots, \varphi_m}$$

vgl. Baum-Spiel

VAL-Spiel

Definition

- Spieler $\exists$ gewinnt:
  1. Parie endet mit $\quad a_1, \dots, a_{|\Sigma|}$
  2. Partie hat ν-Linie

- Spieler $\forall$ gewinnt:
  3. Partie endet mit $\quad b_1, \dots, b_n, \quad n < |\Sigma|$
  4. Partie hat keine ν-Linie

Lemma

$\exists$ *gewinnt VAL* $\quad\Leftrightarrow\quad$ $\exists$ *gewinnt* $MC(t_\Sigma, O\varphi)$

vgl. Baum-Spiel

Sat-Spiele

Lemma

$$\begin{aligned}
\varphi \textit{ erfüllbar} &\Leftrightarrow \quad \textit{es gibt einen Pfad } w \textit{ in } \Sigma^\omega : w \models \varphi \\
&\Leftrightarrow \quad \textit{es gibt einen Pfad } w \textit{ in } \Sigma^\omega : w \not\models \neg\varphi \\
&\Leftrightarrow \quad \neg\varphi \textit{ nicht allgemeingültig} \\
&\Leftrightarrow \quad \forall \textit{ gewinnt } VAL(\neg\varphi)
\end{aligned}$$

SAT-Spiele:

- Negation: $\vee \Leftrightarrow \wedge$, $\quad \mu X \Leftrightarrow \nu X$
- *VAL*: $\forall$ sucht Gegenbsp $\quad \Rightarrow \quad$ *SAT*: $\exists$ sucht Model
- Strategie von $\forall$ in *VAL* $\Rightarrow$ Strategie von $\exists$ in *SAT*
- *VAL*: $\exists$ wählt gleichzeitig $\quad \Rightarrow \quad$ *SAT*: $\forall$ wählt gleichzeitig

Automaten zum Erkennen von ν-Lines

Nicht-det. Büchi Automat

Definition (NBA)

Ein NBA $\mathcal{A} = (Q, A, \delta, q_0, F)$ besteht aus

- Zustände Q
- Alphabet A
- Transitionen $\delta \subseteq Q \times A \times Q$
- Startzustand q_0
- Endzustände $F \subseteq Q$

- Lauf auf $w \in A^\omega$ ist $r = q_0\, q_1 \cdots \in Q^\omega$ gemäß δ.
- $\mathcal{A}$ akzeptiert $w \Leftrightarrow$ ex. Lauf mit $q \in F$ tritt ∞-oft auf.

Problemstellung

Zu lösen mit NBA:

- Eingabe: Partie eines *VAL* Spiels
- Ausgabe: Partie hat/hat keine ν-Line

Definition

Partie hat ν-Line $\Leftrightarrow$

- ex. ν-Variable, die ∞-oft auftritt
- ex. keine größere μ-Variable, die ∞-oft auftritt

vgl. Bsp

Idee

Was macht der NBA?

- Spielablauf beobachten und erst einmal nichts tun
- Wenn ν-Var auftritt:
  Raten ob diese ∞-oft auftritt und keine größere μ-Var.
  In dem Fall fokussieren!
- Akzeptieren, falls fokussierte Var ∞-oft auftritt
- Nicht akzeptieren, falls μ-Var größer fokussierte Var auftritt

Lemma

NBA akzeptiert Partie $\Leftrightarrow$ Partie hat ν-Line

ν-Line NBA

Definition

Der NBA $\mathcal{A}(\varphi) = (Q, A, \delta, q_0, F)$ ist def. durch

- $Q :=$ (Sub(φ) $\times$ ν-Vars) $\cup$ $\{\varepsilon\}$
- $A :=$ Regeln des Spiels
- $\delta := \dots$
- $q_0 := \varepsilon$
- $F := \{(Y, Y) \mid$ Y is ν-Var$\}$

ν-Line NBA

- Transition δ besteht aus:

$$
\begin{array}{lccc}
 & \varepsilon & \xrightarrow{\textit{egal}} & \varepsilon \\
\text{X ist } \nu\text{-Var:} & \varepsilon & \xrightarrow{(X)} & (fb(X), X) \\
 & \dots & & \\
 & (\varphi \land \psi, X) & \xrightarrow{LAND_{\varphi\land\psi}} & (\varphi, X) \\
 & (\varphi \land \psi, X) & \xrightarrow{RAND_{\varphi\land\psi}} & (\psi, X) \\
 & (\varphi \land \psi, X) & \xrightarrow{\textit{sonst}} & (\varphi \land \psi, X) \\
 & \dots & & \\
\text{X ist } \mu\text{-Var:} & (X, Y) & \xrightarrow{(X)} & (fb(X), Y), \quad \text{falls } X < Y
\end{array}
$$

Entscheidungsverfahren

NBA - Determinisierung

Warum determinisieren?

- Partieende bei Wiederholung der Konf gewünscht
- $\forall$ wählt in Abhängigkeit des Fokus anderen Pfad
- Annotieren der Konfs möglich

Safra Konstruktion

Safra Konstruktion (NBA $\rightarrow$ det. Muller Automat)

- Zustände = Safra Bäume
- Transition = Erweiterung der Teilmengen-Konstruktion
- Akzeptanzbedingung:
  Ein Knotenname stirbt nicht aus und blinkt ∞-oft

Eigenschaften

- Knoten: $\mathit{Namen} \times 2^Q \times \{\cdot, !\}$
- max. Knotenzahl: $|Q|$
- #Safta-Bäume $= 2^{O(|Q| \cdot log(|Q|))}$

Q ist Menge der NBA Zustände; $|Q| = |\varphi|^2$

Algorithmus

```
C := φ
c := 0
While c < ∞ Begin
  If GB 1 Then akz.
  If GB 3 Then verw.

  C := ∀ wählt Nachfolgerkonf. (nicht-det.)
  c := c + 1
End

If GB 2 Then akz.
If GB 4 Then verw.
```

Algorithmus

```
(C, t) := (φ, t0)
c := 0
While c < Pmax Begin
  If GB 1 Then akz.
  If GB 3 Then verw.

  Nicht-det. merkt sich ∀: (C',␣t')
  If (C, t) = (C',␣t') Then
    Mullerbed.: verw.

  (C, t) := ∀ wählt Nachfolgerkonf. (+ Annotat)
  c := c + 1
End
akz.
```

Algorithmus

```
(C, t) := (φ, t₀)
c := 0
While c < P_max Begin
  If GB 1 Then akz.
  If GB 3 Then verw.

  Nicht-det. merkt sich ∀: (C',␣t')
     N := Knoten von t'
   N := N ∩ Knoten von t
   If (C, t) = (C', t') Then
     If Blitz-Knoten ∈ N Then verw.

  (C, t) := ∀ wählt Nachfolgerkonf. (+ Annotat)
  c := c + 1
End
akz.
```

Abschätzung

Abschätzung:

- C polynomiell in $|\varphi|$: $\quad |C| \leq |Sub(\varphi)| = O(|\varphi|)$
- t polynomiell in $|\varphi|$:
  - $\#$Knoten $\leq |Q| \leq |\varphi|^2$
  - Knotenname, Lampe klar
  - Menge der NBA Zustände $\leq |Sub(\varphi)|^2 = O(|\varphi|^2)$
- $P_{max} \leq 2^{|Sub(\varphi)|} \cdot 2^{O(|Q|)\cdot log(|Q|)} \Rightarrow c \in log(\dots)$
- N in $O(|Q|) = O(|\varphi|^2)$

$\Rightarrow$ mit Savich folgt Alg. in PSPACE

Das war's.