From Büchi Automata to Cyclic and Infinite Proofs
Internship at ITU Copenhagen

Lucca Hirschi

July 10, 2012

Lucca Hirschi
directed by
ENS Lyon

David Baelde
director of
ITU of Copenhagen
Encode Büchi automata as formulas in a proof-theoretical framework with (co)-induction.
Purpose

Encode Büchi automata as formulas in a proof-theoretical framework with (co)-induction.

Logics dealing with infinite proofs, cyclic proofs; mixing inductive and co-inductive formulas; strongly related; well describe Büchi Automata.
Big Picture

1 1 0 0
q₁ q'₁

↓

Common and used: explicit (co)-induction

μLK ⊆ μLK^ω ⊆ μLK^∞
Infinite proofs: satisfies adequacy, impractical

Common and used: explicit (co)-induction

$\muLK \subseteq \muLK^\omega \subseteq \muLK^\infty$
Big Picture

- Infinite proofs: satisfies adequacy, impractical
- Common and used: explicit (co)-induction

\[ \mu_{LK} \subset \mu_{LK}^\omega \subset \mu_{LK}^\infty \]

- Cyclic proofs
Outline

1. Introduction
2. Büchi Automata
3. $\mu$LK
4. $\mu$LK$^\omega$ and $\mu$LK$^\infty$
5. Conclusion
Büchi Automata

Definition (Büchi Automata)

A Büchi automaton is a quintuple $\mathcal{A} = (Q, \Sigma, \delta, Q_I, Q_F)$, where
- $Q$ is a finite set (the states);
- $\Sigma$ is an alphabet;
- $\delta : Q \times \Sigma \to \mathcal{P}(Q)$ the nondeterministic transition function;
- $Q_I \subseteq Q$ the initial states and $Q_F \subseteq Q$ the final states.
Büchi Automata

**Definition (Büchi Automata)**

A *Büchi automaton* is a quintuple $\mathcal{A} = (Q, \Sigma, \delta, Q_I, Q_F)$, where

- $Q$ is a finite set (the states);
- $\Sigma$ is an alphabet;
- $\delta : Q \times \Sigma \rightarrow \mathcal{P}(Q)$ the nondeterministic transition function;
- $Q_I \subseteq Q$ the initial states and $Q_F \subseteq Q$ the final states.

**Definition (Acceptance condition)**

- A run $\alpha$ on a word is **accepting** by an automaton $\iff$ $\alpha$ visits a final state infinitely often;
- A word is **recognized** by an automaton $\iff$ there exists an accepting run on it.
An Example of a Büchi Automaton

$L(\mathcal{A}) = (0^*1)^\omega$
An Example of a Büchi Automaton

\[ L(A) = (0^*1)^\omega \]

\[ \vdash T_A(1^\omega) \]
An Example of a Büchi Automaton

\[ L(\mathcal{A}) = (0^*1)^\omega \]

“Proof”

By reading the word \(1^\omega\), I can build step by step an accepting run in \(\mathcal{A}\):

“From state \(q_1\), I read 1 and jump to \(q_1\) and so on so forth.”
An Example of a Büchi Automaton

\[ \vdash [q_1] 1^\omega \]
An Example of a Büchi Automaton

\[ \vdash \lbrack q_1 \rbrack 1^\omega \]

\[ \vdash \lbrack q_1 \rbrack 1 :: 1^\omega \]

\[ \vdash \exists tl \ (1 :: 1^\omega = 1 :: tl \land \lbrack q_1 \rbrack tl) \lor (1 :: 1^\omega = 0 :: tl \land \lbrack q'_1 \rbrack tl) \]

\[ \vdash \lbrack q_1 \rbrack 1 :: 1^\omega \]
Goals

- **Adequacy**: \( w \in L(A) \iff \vdash [A][w] \)
Goals

- **Adequacy**: $w \in \mathcal{L}(A) \iff \vdash [A][w]$ and a link between computations in Büchi automata and proofs of their properties;
Goals

- **Adequacy:** \( w \in \mathcal{L}(\mathcal{A}) \iff \vdash \langle \mathcal{A} \rangle[w] \) and a link between computations in Büchi automata and proofs of their properties;

- **Soundness and completeness of inclusion:** our main problem is the inclusion. We must show that

\[
\| A_1 \| x \vdash \| A_2 \| x \iff \mathcal{L}(\mathcal{A}_1) \subseteq \mathcal{L}(\mathcal{A}_2).
\]
Goals

- **Adequacy:** $w \in \mathcal{L}(A) \iff \vdash \|A\|_w$ and a link between computations in Büchi automata and proofs of their properties;

- **Soundness and completeness of inclusion:** our main problem is the inclusion. We must show that
  \[ \|A_1\|_x \vdash \|A_2\|_x \iff \mathcal{L}(A_1) \subseteq \mathcal{L}(A_2). \]
  Proof of inclusion $\leadsto$ inclusion and a certificate;
Goals

- **Adequacy:** \( w \in L(A) \iff \vdash \llbracket A \rrbracket[w] \) and a link between computations in Büchi automata and proofs of their properties;

- **Soundness and completeness of inclusion:** our main problem is the inclusion. We must show that

  \[
  \llbracket A_1 \rrbracket x \vdash \llbracket A_2 \rrbracket x \iff L(A_1) \subseteq L(A_2).
  \]

  Proof of inclusion \( \sim \) inclusion and a certificate;

- **Usable and generic logic:** properties over automata are used in a wider context.
Outline

1. Introduction
2. Büchi Automata
3. \( \mu LK \)
4. \( \mu LK^\omega \) and \( \mu LK^\infty \)
5. Conclusion
Definition (Formula of $\mu$LK)

$$P ::= \top | \bot$$

$$| \exists x. \ P \ | \ \forall x. \ P \quad \quad x \in V$$

$$| P \land P \ | \ P \lor P \ | \ P \Rightarrow P$$

$$| s = t \quad \quad t, s \text{ some terms}$$
\( \mu LK \)

**Definition (Formula of \( \mu LK \))**

\[
P ::= \top | \bot \\
| \exists x. \ P | \forall x. \ P \quad \text{\( x \in \mathcal{V} \)} \\
| P \land P | P \lor P | P \Rightarrow P \\
| s = t \quad \text{\( t, s \) some terms} \\
| p \quad \text{\( p \in \mathcal{V}_f \)} \\
| \mu(p.\lambda x_1 \ldots \lambda x_n. \ P) \ t_1 \ldots t_n \quad \text{\( p \in \mathcal{V}_f, \ t_i \) a term} \\
| \nu(p.\lambda x_1 \ldots \lambda x_n. \ P) \ t_1 \ldots t_n \quad \text{\( p \in \mathcal{V}_f, \ t_i \) a term}
\]
Definition (Formula of $\mu$LK)

$$
P ::= \top \mid \bot \mid \exists x. P \mid \forall x. P \quad x \in V \\
    \mid P \land P \mid P \lor P \mid P \Rightarrow P \\
    \mid s = t \quad t, s \text{ some terms} \\
    \mid p \quad p \in V_f \\
    \mid \mu(\lambda p. \lambda x_1. \ldots . \lambda x_n. P) \ t_1 \ldots t_n \quad p \in V_f, \ t_i \text{ a term} \\
    \mid \nu(\lambda p. \lambda x_1. \ldots . \lambda x_n. P) \ t_1 \ldots t_n \quad p \in V_f, \ t_i \text{ a term}
$$

$$
N = \mu B_{\text{nat}} = \mu (\lambda p_n. \lambda x. x = 0 \lor (\exists y \ x = s(y) \land p_n \ y)) \\
S = \nu B_{\text{stream}} = \nu (\lambda p_s. \lambda w. \exists w' \exists n \ w = n : w' \land N n \land p_s \ w')
$$
Rules of $\mu$LK

Sequent calculus:

- identity group: $Ax$, cut, $\equiv R$, $\equiv L$;
- logical group: $\top$, $\bot$, $\land L_i$, $\land R$, $\lor L$, $\lor R_i$, $\Rightarrow L$, $\Rightarrow R$, $\forall L$, $\forall R$, $\exists L$, $\exists R$;
- structural group: $WL$, $WR$ (weak), $CL$, $CR$ (contraction).
Rules of $\mu$LK

Sequent calculus:

- identity group: $\text{Ax}$, cut, $= R, = L$;
- logical group: $\top, \bot, \land L_i, \land R, \lor L, \lor R_i, \Rightarrow L, \Rightarrow R, \forall L, \forall R, \exists L, \exists R$;
- structural group: $WL, WR$ (weak), $CI, CR$ (contraction).

+ explicit (co)-induction:

```
Γ ⊢ B(μB) t  Γ ⊢ μB t
------------------------  μR
Γ ⊢ μB t

Γ, S t ⊢ P  BS x ⊢ S x
------------------------  μL
Γ, μB t ⊢ P

Γ ⊢ St  St ⊢ BSt
------------------------  νR
Γ ⊢ νB t

Γ ⊢ B(νB) t ⊢ P
------------------------  νL
Γ, νB t ⊢ P
```

Lucca Hirschi
\[
\frac{\Gamma \vdash B(\mu B) \, t}{\Gamma \vdash \mu B \, t} \quad \text{\(\mu R\)}
\]

\[
\frac{\Gamma, S \, t \vdash P \quad BS \, x \vdash S \, x}{\Gamma, \mu B \, t \vdash P} \quad \text{\(\mu L\)}
\]

\[
\frac{\Phi_0 \text{ or } \Phi_n}{\Gamma \vdash t = 0 \lor \exists y \, t = s(y) \land \mu B_{\text{nat}} \, y} \quad \text{\(\mu R\)}
\]

\[
\frac{\Gamma \vdash \mu B_{\text{nat}} \, t}{\Gamma \vdash \mu B_{\text{nat}} \, t} \quad \text{\(\mu R\)}
\]
\[\Gamma \vdash B(\mu B) \; t \quad \mu R \quad \mu L\]

\[\Gamma, \; S \; t \vdash P \quad BS \; x \vdash S \; x\]

\[\Phi_0 \text{ or } \Phi_n\]

\[\Gamma \vdash t = 0 \lor \exists y \; t = s(y) \land \mu B_{\text{nat}} \; y \quad \mu R\]

\[\prod \quad S \; t \vdash P\]

\[\begin{array}{c}
\psi_0 \\
\vdash S \; 0 \\
S \; x \vdash S \; (s(x)) \\
\end{array}\]

\[x = 0 \lor \exists y \; x = s(y) \land S \; y \vdash S \; x\]

\[\forall L, \; (\exists L), = L\]

\[\mu B_{\text{nat}} \; t \vdash P \quad \mu L\]
\[ \frac{\Gamma \vdash S t \quad S t \vdash B S t}{\Gamma \vdash \nu B \ t} \quad \nu R \]
\[ \frac{\Gamma, B(\nu B) t \vdash P}{\Gamma, \nu B t \vdash P} \quad \nu L \]

\[ \frac{\Gamma \vdash S t \quad \exists t' \exists n \ t = n : t' \land N n \land S t' \vdash S t}{\Gamma \vdash \nu B_{\text{stream}} t} \quad \nu R \]
\[
\begin{align*}
\Gamma \vdash S \Gamma, B(\nu B) t \vdash P & \quad \nu R \\
\Gamma \vdash \nu B t & \quad \Gamma, \nu B t \vdash P & \quad \nu L
\end{align*}
\]

\[
\begin{align*}
\Gamma \vdash S t & \quad \exists t' \exists n t = n : t' \land N n \land S t' \vdash S t & \quad \nu R \\
\Gamma \vdash \nu B_{\text{stream}} t & \quad \Gamma \vdash S t
\end{align*}
\]

\[
\begin{align*}
t = n :: t' \land \nu B_{\text{stream}} t' & \quad \nu L \\
\nu B_{\text{stream}} t \vdash P
\end{align*}
\]
\( \mu LK \) vs. Büchi automata

\( [q_1] = \nu(\lambda q_1 . \lambda w. \exists w' \\
(w = 1 :: w' \land q_1 w') \lor \\
(w = 0 :: w' \land [q_1'] w') 
) \)

\( [q_1'] = \mu(\lambda q_1' . \lambda w. \exists w' \\
(w = 1 :: w' \land [q_1] w') \lor \\
(w = 0 :: w' \land q_1' w') 
) \)
\[ \mu \text{LK vs. Büchi automata} \]

Which S? Why?

\[ \vdash_{\mu \text{LK}} [q_1] \mathbb{1}^\omega \quad \nu R \]

\[ [q_1] = \nu (\lambda q_1. \lambda w. \exists w' \\
(w = 1 :: w' \land q_1 w') \lor \\
(w = 0 :: w' \land [q_1'] w') \) \]

\[ [q_1'] = \mu (\lambda q_1'. \lambda w. \exists w' \\
(w = 1 :: w' \land [q_1] w') \lor \\
(w = 0 :: w' \land q_1' w') \) \]
\[\muLK \text{ vs. Büchi automata}\]

Which S? Why?

\[\vdash_{\muLK} [q_1] 1^\omega \quad \nu R\]

\[([q_1] = \nu(\lambda q_1. \lambda w. \exists w' (w = 1 :: w' \land q_1 \ w') \lor (w = 0 :: w' \land [q_1'] w')))
\]

\[([q_1'] = \mu(\lambda q_1'. \lambda w. \exists w' (w = 1 :: w' \land [q_1] w') \lor (w = 0 :: w' \land q_1' \ w')) \lor R, \lor R_2 \quad \nu R')\]

\[\vdash [q_1] 1^\omega \quad \exists w' (1^\omega = 1 :: w' \land [q_1] w') \lor (1^\omega = 0 :: w' \land [q_1'] w') \quad \nu R'\]
\[ \mu LK \] vs. Büchi automata

Which S? Why?

\[ \vdash_{\mu LK} \llbracket q_1 \rrbracket \omega \]

\[ \forall R \]

\[ \llbracket q_1 \rrbracket = \forall (\lambda q_1. \lambda w. \exists w' \left( w = 1 : w' \land q_1 \cdot w' \right) \lor \left( w = 0 : w' \land [q'_1] w' \right) \} \]

\[ \llbracket q'_1 \rrbracket = \mu (\lambda q'_1. \lambda w. \exists w' \left( w = 1 : w' \land [q_1] w' \right) \lor \left( w = 0 : w' \land q'_1 \cdot w' \right) \} \]

\[ \leadsto \alpha \]

\[ \vdash \exists w' \left( \omega_1 = 1 : w' \land [q_1] \omega \right) \lor \left( \omega_1 = 0 : w' \land [q'_1] \omega \right) \]

\[ \exists R, \forall R' \]

\[ \vdash \llbracket q_1 \rrbracket \omega : \alpha \]
\[ [q_1] = \nu(\lambda q_1. \lambda w. \exists w' \\
( w = 1 :: w' \land q_1 w') \lor \\
( w = 0 :: w' \land [q'_1] w') ) \]

\[ [q'_1] = \mu(\lambda q'_1. \lambda w. \exists w' \\
( w = 1 :: w' \land [q_1] w') \lor \\
( w = 0 :: w' \land q'_1 w') ) \]

Which S? Why?

\[ \vdash \muLK [q_1] 1^\omega \quad \nu R \]

\[ \vdash \exists w' \\
( 1^\omega = 1 :: w' \land [q_1] w') \lor \\
( 1^\omega = 0 :: w' \land [q'_1] w') \\
\vdash \exists R, \lor R_2 \\
\vdash [q_1] 1^\omega : \alpha \]
Outline

1. Introduction
2. Büchi Automata
3. $\mu$LK
4. $\mu$LK$^\omega$ and $\mu$LK$^\infty$
5. Conclusion
Explicit (co)-induction rules are replaced by some cycles or infinite branches.
Explicit (co)-induction rules \( \leadsto \) replaced by some cycles or infinite branches.

\[
\mu B_{\text{even}} = \mu (\lambda p_n. \lambda x. x = 0 \lor (\exists y \ x = s(s(y)) \land p_n y))
\]

\[
\begin{align*}
\frac{t = 0 \vdash \mu B_{\text{nat}} t}{\mu B_{\text{even}} t' \vdash \mu B_{\text{nat}} (s(s(t')))} & \quad \frac{\mu B_{\text{even}} t' \vdash \mu B_{\text{nat}} t}{t = s(s(t')) \land \mu B_{\text{even}} t' \vdash \mu B_{\text{nat}} t} \\
\mu B_{\text{even}} t \vdash \mu B_{\text{nat}} t & \quad \mu L'
\end{align*}
\]
Explicit (co)-induction rules \(\leadsto\) replaced by some cycles or infinite branches.

\[
\mu B_{\text{even}} = \mu \left( \lambda p_n \ldotp \lambda x. \ x = 0 \lor (\exists y \ x = s(s(y)) \land p_n \ y) \right)
\]

\[
\begin{align*}
\mu B_{\text{even}} \ t' & \vdash \mu B_{\text{nat}} \ t' \\
\mu B_{\text{even}} \ t' & \vdash \mu B_{\text{nat}} \ (s(t')) \\
\mu B_{\text{even}} \ t' & \vdash \mu B_{\text{nat}} \ (s(s(t')))) \\
\end{align*}
\]

\[
\begin{align*}
t = 0 \vdash \mu B_{\text{nat}} \ t \\
t = s(s(t')) \land \mu B_{\text{even}} \ t' \vdash \mu B_{\text{nat}} \ t \\
\mu B_{\text{even}} \ t \vdash \mu B_{\text{nat}} \ t \\
\end{align*}
\]
Explicit (co)-induction rules \( \sim \) replaced by some cycles or infinite branches.

\[
\mu B_{\text{even}} = \mu (\lambda p_n . \lambda x. x = 0 \lor (\exists y \ x = s(s(y)) \land p_n y))
\]

\[
\mu LK^\omega \quad \mu LK^\infty
\]

\[
\mu B_{\text{even}} \vdash \mu B_{\text{nat}} t' \\
\mu B_{\text{even}} \vdash \mu B_{\text{nat}} (s(t')) \\
\mu B_{\text{even}} \vdash \mu B_{\text{nat}} (s(s(t')))
\]

\[
t = 0 \vdash \mu B_{\text{nat}} t \\
t = s(s(t')) \land \mu B_{\text{even}} t' \vdash \mu B_{\text{nat}} t
\]

\[
\mu B_{\text{even}} t \vdash \mu B_{\text{nat}} t : \alpha \quad \mu L'
\]
Guard Condition

Litterature:

1. Brotherstone: No co-inductive formula; “infinite descent”; [Bro06]
Guard Condition

Litterature:

1. Brotherstone: No co-inductive formula; “infinite descent”; [Bro06]
2. Santocanale: No cut rule; inductive and co-inductive formula; [San02]
Guard Condition

Litterature:

1. Brotherstone: No co-inductive formula; “infinite descent”; [Bro06]
2. Santocanane: No cut rule; inductive and co-inductive formula; [San02]

First bug

\[
\begin{align*}
P &= \mu(\lambda p. \nu(\lambda q. p)) \\
Q &= \nu(\lambda q. P)
\end{align*}
\]

\[
\frac{\vdash Q}{\vdash P} \quad \frac{\vdash Q : \alpha}{\vdash P} \quad \frac{\vdash P}{\vdash Q} \quad \frac{\vdash P}{\vdash \bot} \\
\frac{\vdash \bot}{\vdash P} \quad \frac{\vdash \bot}{\vdash \bot} \quad \frac{\vdash \bot}{\vdash \bot} \\
\frac{\vdash \alpha}{\vdash \mu R} \quad \frac{\vdash \nu R}{\vdash \mu R} \quad \frac{\vdash \tau}{\vdash \nu L} \quad \frac{\vdash \tau}{\vdash \mu L}
\]

\text{cut}
Guard Condition - interleaved (co)-inductive formulas

**First bug**

\[
\begin{align*}
  P &= \mu(\lambda p. \nu(\lambda q. p)) \\
  Q &= \nu(\lambda q. P)
\end{align*}
\]

\[
\begin{array}{l}
\frac{}{\vdash Q} \\
\frac{}{\vdash P} \\
\frac{\vdash Q}{\vdash Q : \alpha} \\
\frac{\vdash P}{\vdash P} \\
\frac{}{\vdash \bot} \\
\frac{\vdash \bot}{\vdash \bot} \\
\frac{}{\vdash \bot}
\end{array}
\]

\[
\begin{array}{l}
\frac{\vdash \bot}{\vdash \bot} \\
\frac{\vdash \bot}{\vdash \bot} \\
\frac{}{\vdash \bot}
\end{array}
\]

**Fix**

\[
\begin{align*}
  P &=_{\mu} Q \\
  Q &=_{\nu} P \\
  P &> Q
\end{align*}
\]
Guard Condition - interleaved (co)-inductive formulas

First bug

\[
\begin{align*}
P &= \mu(\lambda q. \nu(\lambda q. p)) \\
Q &= \nu(\lambda q. P)
\end{align*}
\]

\[
\begin{array}{c}
\frac{}{\vdash \alpha} \\
\frac{\vdash Q}{\vdash P} \\
\frac{\vdash Q : \alpha}{\vdash P} \\
\frac{\vdash P}{\vdash \bot}
\end{array}
\quad
\begin{array}{c}
\frac{}{\vdash \tau} \\
\frac{\vdash P}{\vdash \bot} \\
\frac{\vdash Q}{\vdash \bot}
\end{array}
\]

\[
\begin{array}{c}
\vdash \mu R \\
\vdash \nu R \\
\vdash \mu L \\
\vdash \nu L
\end{array}
\]

\[
\begin{array}{c}
\text{cut}
\end{array}
\]

Fix

\[
\begin{align*}
P &= \nu Q \\
Q &= \mu P
\end{align*}
\]

\[
\begin{array}{c}
\frac{}{\vdash \alpha} \\
\frac{\vdash Q}{\vdash P} \\
\frac{\vdash Q : \alpha}{\vdash P} \\
\frac{\vdash P}{\vdash \bot}
\end{array}
\quad
\begin{array}{c}
\frac{}{\vdash \tau} \\
\frac{\vdash P}{\vdash \bot} \\
\frac{\vdash Q}{\vdash \bot}
\end{array}
\]

\[
\begin{array}{c}
\vdash \nu R_Q \\
\vdash \nu L_Q \\
\vdash \mu R_P \\
\vdash \mu L_P
\end{array}
\]

\[
\begin{array}{c}
\text{cut}
\end{array}
\]

Lucca Hirschi
Guard Condition - interleaved (co)-inductive formulas

First bug

\[
\begin{align*}
P &= \mu(\lambda p. \nu(\lambda q. p)) \\
Q &= \nu(\lambda q. P)
\end{align*}
\]

\[
\begin{array}{c}
\vdash Q \\
\vdash P \\
\vdash Q : \alpha \\
\vdash P \\
\vdash \bot
\end{array}
\]

\[
\begin{array}{c}
\vdash Q \\
\vdash P \\
\vdash Q : \alpha \\
\vdash P \\
\vdash \bot \\
\vdash \bot
\end{array}
\]

\[
\begin{array}{c}
\vdash \tau \\
\vdash Q \\
\vdash \bot \\
\vdash P \\
\vdash \bot
\end{array}
\]

\[
\begin{array}{c}
\vdash \tau \\
\vdash Q \\
\vdash \bot \\
\vdash P \\
\vdash \bot
\end{array}
\]

\[
\begin{array}{c}
\vdash \alpha \\
\vdash Q \\
\vdash P \\
\vdash Q : \alpha \\
\vdash P \\
\vdash \bot \\
\vdash \bot
\end{array}
\]

\[
\begin{array}{c}
\vdash \alpha \\
\vdash Q \\
\vdash P \\
\vdash Q : \alpha \\
\vdash P \\
\vdash \bot \\
\vdash \bot
\end{array}
\]

\[
\begin{array}{c}
\vdash \tau \\
\vdash Q \\
\vdash \bot \\
\vdash P \\
\vdash \bot
\end{array}
\]

\[
\begin{array}{c}
\vdash \tau \\
\vdash Q \\
\vdash \bot \\
\vdash P \\
\vdash \bot
\end{array}
\]

Fix

\[
\begin{align*}
P &= \mu, Q \\
Q &= \nu, P \\
P &> Q
\end{align*}
\]

\[
\begin{array}{c}
\vdash Q \\
\vdash P \\
\vdash Q : \alpha \\
\vdash P \\
\vdash \bot
\end{array}
\]

\[
\begin{array}{c}
\vdash Q \\
\vdash P \\
\vdash Q : \alpha \\
\vdash P \\
\vdash \bot \\
\vdash \bot
\end{array}
\]

\[
\begin{array}{c}
\vdash \tau \\
\vdash Q \\
\vdash \bot \\
\vdash P \\
\vdash \bot
\end{array}
\]

\[
\begin{array}{c}
\vdash \tau \\
\vdash Q \\
\vdash \bot \\
\vdash P \\
\vdash \bot
\end{array}
\]

\[
\begin{array}{c}
\vdash \alpha \\
\vdash Q \\
\vdash P \\
\vdash Q : \alpha \\
\vdash P \\
\vdash \bot \\
\vdash \bot
\end{array}
\]

\[
\begin{array}{c}
\vdash \alpha \\
\vdash Q \\
\vdash P \\
\vdash Q : \alpha \\
\vdash P \\
\vdash \bot \\
\vdash \bot
\end{array}
\]

\[
\begin{array}{c}
\vdash \tau \\
\vdash Q \\
\vdash \bot \\
\vdash P \\
\vdash \bot
\end{array}
\]

\[
\begin{array}{c}
\vdash \tau \\
\vdash Q \\
\vdash \bot \\
\vdash P \\
\vdash \bot
\end{array}
\]
“Definition”: table of (co)-induction

\[(Q, \, \epsilon, \supseteq, \prec)\]

- \(Q\): names of (co)-inductive formulas (defined atoms);
- \(\epsilon\): \(Q \rightarrow \{\mu; \nu\}\);
- \(P \supseteq A\): \(A\) is the unfolding of \(P \in Q\);
- \(\prec\): Who is on the top of who?
**“Definition”: table of (co)-induction**

\[(Q, \epsilon, \geq, <)\]

- **\(Q\):** names of (co)-inductive formulas (defined atoms);
- **\(\epsilon: Q \rightarrow \{\mu; \nu\}\);**
- **\(P \geq A\):** \(A\) is the unfolding of \(P \in Q\);
- **\(<\):** Who is on the top of who?

**Second bug**

\[
\begin{align*}
\text{Nat} & \geq B_{\text{nat}} \text{Nat} \quad \epsilon(\text{Nat}) = \mu \\
B_{\text{nat}} & = \lambda p_n. \lambda n. n = 0 \lor \exists n' \ n = s(n') \land p_n \ n'
\end{align*}
\]

\[
\text{Nat} \vdash \bot : \alpha \mu L
\]
“Definition”: table of (co)-induction

\[(Q, \epsilon, \geq, <)\]

- \(Q\): names of (co)-inductive formulas (defined atoms);
- \(\epsilon\): \(Q \rightarrow \{\mu; \nu\}\);
- \(P \geq A\): \(A\) is the unfolding of \(P \in Q\);
- \(<\): Who is on the top of who?

Second bug

\[
\begin{align*}
\text{Nat} & \geq B_{\text{nat}} \text{Nat} & \epsilon(\text{Nat}) = \mu \\
B_{\text{nat}} & = \lambda p_n. \lambda n. \ n = 0 \lor \exists n' \ n = s(n') \land p_n \ n'
\end{align*}
\]

\[
\begin{array}{c}
\frac{B_{\text{nat}} \text{Nat} \ t \vdash B_{\text{nat}} \text{Nat} \ t}{B_{\text{nat}} \text{Nat} \ t \vdash \text{Nat} \ t} \ \text{Ax} \\
\frac{\mu R}{\text{Nat} \ t \vdash \bot} \ \uparrow \alpha \\
\frac{B_{\text{nat}} \text{Nat} \ t \vdash \bot}{\text{Nat} \ t \vdash \bot} \ \text{cut} \\
\frac{B_{\text{nat}} \text{Nat} \ t \vdash \bot}{\text{Nat} \ t \vdash \bot} \ : \ \alpha \ \mu L
\end{array}
\]
Guard Condition - observation

\[
\begin{align*}
\uparrow \alpha \\
\frac{}{s_5 : \text{Even } t' \vdash \text{Nat } t'} \\
\frac{}{s_4 : \text{Even } t' \vdash \text{Nat } (s(t'))} \\
\frac{}{s_3 : \text{Even } t' \vdash \text{Nat } (s(s(t')))} \\
\frac{t = 0 \vdash \text{Nat } t}{s_2 : t = s(s(t')) \land \text{Even } t' \vdash \text{Nat } t} \\
\frac{}{s_1 : (A =) \text{Even } t \vdash \text{Nat } t : \alpha}
\end{align*}
\]
Guard Condition - observation

\[
\begin{align*}
\uparrow \alpha \\
{s_5} & : \text{Even } t' \vdash \text{Nat } t' \\
{s_4} & : \text{Even } t' \vdash \text{Nat } (s(t')) \\
{s_3} & : \text{Even } t' \vdash \text{Nat } (s(s(t')))
\end{align*}
\]

\[
\begin{align*}
 t = 0 & \vdash \text{Nat } t \\
{s_2} & : t = s(s(t')) \land \text{Even } t' \vdash \text{Nat } t \\
{s_1} & : (A =) \text{Even } t \vdash \text{Nat } t : \alpha
\end{align*}
\]

\[
O_A(\alpha) = (\mu \mathcal{L}, \text{Even})
\]
**Guard Condition - observation**

\[
\begin{align*}
\uparrow \alpha \\
& \vdash s_5 : \text{Even } t' \vdash \text{Nat } t' \\
& \vdash s_4 : \text{Even } t' \vdash \text{Nat } (s(t')) \\
& \vdash s_3 : \text{Even } t' \vdash \text{Nat } (s(s(t'))) \\
& \vdash t = 0 \vdash \text{Nat } t \\
& \vdash s_2 : t = s(s(t')) \land \text{Even } t' \vdash \text{Nat } t \\
& \vdash s_1 : (A = )\text{Even } t \vdash \text{Nat } t : \alpha
\end{align*}
\]

\[O_A(\alpha) = (\mu L, \text{Even})\]

"Definition": Observations

- The trace of \(A \in s_0\) in the branch \(s_0, s_1, \ldots\) is a serie of formulas \(A_0, A_1, \ldots\) such that:
  - \(A_i \in s_i\) (on the same side);
  - if \(A_i\) is active in the conclusion \(s_i\) then \(A_{i+1}\) is active in the premise of \(s_{i+1}\).

- The observation of a formula in a branch is the serie of \((r, A)\) where \(r\) is a (co)-inductive rules applied to \(A\) appearing in the trace.
Guard Condition

\[ B_{\text{nat}} \text{Nat } t \vdash B_{\text{nat}} \text{Nat } t \]

\[ B_{\text{nat}} \text{Nat } t \vdash \text{Nat } t \]

\[ \mu R \]

\[ \text{Nat } t \vdash \text{Nat } t \]

\[ \text{cut} \]

\[ B_{\text{nat}} \text{Nat } t \vdash \bot \]

\[ \text{Nat } t \vdash \bot : \alpha \]

\[ \mu L \]
Guard Condition

\[ \frac{B_{\text{nat}} \text{Nat} t \vdash B_{\text{nat}} \text{Nat} t}{B_{\text{nat}} \text{Nat} t \vdash \text{Nat} t} \quad \frac{\text{Ax}}{\mu R} \quad \frac{\uparrow \alpha}{\text{Nat} t \vdash \text{Nat} t} \quad \text{cut} \]

\[ \frac{B_{\text{nat}} \text{Nat} t \vdash \bot}{\text{Nat} t \vdash \bot} : \alpha \quad \frac{\mu L}{\mu \text{L}} \]
\[ B_{\text{nat}} \text{Nat} t \vdash B_{\text{nat}} \text{Nat} t \quad \text{Ax} \]
\[ \mu R \]
\[ \uparrow \alpha \]
\[ \text{Nat} t \vdash \text{Nat} t \]
\[ \text{cut} \]
\[ B_{\text{nat}} \text{Nat} t \vdash \bot \quad \mu L \]
\[ \text{Nat} t \vdash \bot : \alpha \]

"Definition": Refinement of Guard Condition

A proof is valid \iff each infinite brach is either inductive or co-inductive.

- **inductive branch**: there is an observation \( o \) on the left such that that \( \epsilon \left( \max_{(r,n) \in \text{Inf}(o)} \{n\} \right) = \mu \);
“Definition”: Refinement of Guard Condition

A proof is valid $\iff$ each infinite brach is either inductive or co-inductive.

- **inductive branch**: there is an observation $o$ on the left such that $\epsilon \left( \max_{(r,n) \in \text{Inf}(o)} \{n\} \right) = \mu$;
- **co-inductive branch**: there is an observation $o$ on the right such that $\epsilon \left( \max_{(r,n) \in \text{Inf}(o)} \{n\} \right) = \nu$. 

\[
\begin{array}{c}
B_{\text{nat}} \text{Nat } t \vdash B_{\text{nat}} \text{Nat } t \quad \text{Ax} \\
\mu R \\
B_{\text{nat}} \text{Nat } t \vdash \text{Nat } t \\
\mu L \\
\end{array}
\]

\[
\begin{array}{c}
\mu \text{nat } \text{Nat } t \vdash \mu \text{nat } \text{Nat } t \\
\text{cut} \\
\mu \text{nat } \text{Nat } t \vdash \perp \\
\end{array}
\]

\[
\begin{array}{c}
\mu \text{nat } \text{Nat } t \vdash \perp : \alpha \\
\mu L \\
\end{array}
\]
Guard Condition

\[ \mu L K^\infty \]

- Bijection between observations of \([q] t\) and runs starting with \(q\);
\[ \mu L K^\infty \]

- Bijection between observations of \([q_t]\) and runs starting with \(q\);
- Completeness and soundness of acceptance;
Bijection between observations of \([q] t\) and runs starting with \(q\);
- completeness and soundness of acceptance;
- completeness and soundness of inclusion.
“Definition”: Refinement of Guard Condition

A proof is valid $\iff$ each cycle is either inductive or co-inductive.

- **inductive cycle**: there is an observation $o$ on the left such that 
  $\epsilon\left(\max_{(r,n)\in o}\{n\}\right) = \mu$;
“Definition”: Refinement of Guard Condition

A proof is valid $\iff$ each cycle is either inductive or co-inductive.

- **inductive cycle**: there is an observation $o$ on the left such that
  $\epsilon \left( \max_{(r,n) \in o} \{ n \} \right) = \mu$;

- **co-inductive cycle**: there is an observation $o$ on the right such that
  $\epsilon \left( \max_{(r,n) \in o} \{ n \} \right) = \nu$. 
Unexpected bug

\[ L(L) \not\subseteq L(A) \]

\[(0::0::1::1)^\omega \not\subseteq L(A)\]
Unexpected bug

\[ \mathcal{L}(L) \not\subseteq \mathcal{L}(A) \quad (0 :: 0 :: 1 :: 1)^\omega \not\subseteq \mathcal{L}(A) \]

Validity condition holds but does not respect the Büchi automata semantics.
Unexpected bug

\[ L(L) \not\subseteq L(A) \quad (0 :: 0 :: 1 :: 1) ^ \omega \not\subseteq L(A) \]

Validity condition holds but does not respect the Büchi automata semantics. Traces are broken.
A few counter-examples later

“Definition”: Refinement\(^2\) of Guard Condition

A proof is valid \(\iff\) each cycle is either inductive or co-inductive.

- **Inductive cycle**: there is an observation \(o = o_1, o_2, \ldots o_p\) on the left such that:
  - \(o_1 = o_p\);
  - \(\max_{(r,n) \in o} \{n\} = n_1\) and \(\epsilon(n_1) = \mu\);

- **Co-inductive cycle**: there is an observation \(o = o_1, o_2, \ldots o_p\) on the right such that:
  - \(o_1 = o_p\);
  - \(\max_{(r,n) \in o} \{n\} = n_1\) and \(\epsilon(n_1) = \nu\).
Outline

1. Introduction
2. Büchi Automata
3. $\mu LK$
4. $\mu LK^\omega$ and $\mu LK^\infty$
5. Conclusion
Conclusion

$$\mu LK = \mu LK^\omega \subseteq \mu LK^\infty$$

1. New logic $\mu LK^\omega$ and $\mu LK^\infty$ supporting mutual inductive and coinductive definitions with implicit (co)induction;
\[ \mu LK = \mu LK^\omega \subseteq \mu LK^\infty \]

1. New logic \( \mu LK^\omega \) and \( \mu LK^\infty \) supporting mutual inductive and coinductive definitions with implicit (co)induction;

2. they are strongly related with \( \mu LK \): we can translate back and forth between \( \mu LK \) proofs and cyclic proofs;
Conclusion

\[ \mu LK = \mu LK^\omega \subseteq \mu LK^\infty \]

1. New logic \( \mu LK^\omega \) and \( \mu LK^\infty \) supporting mutual inductive and coinductive definitions with implicit (co)induction;

2. they are strongly related with \( \mu LK \): we can translate back and forth between \( \mu LK \) proofs and cyclic proofs;

3. they mirror closely the mathematical structure of Büchi automata and their computations: adequacy;
\[ \mu LK = \mu LK^\omega \subseteq \mu LK^\infty \]

1. New logic \(\mu LK^\omega\) and \(\mu LK^\infty\) supporting mutual inductive and coinductive definitions with implicit (co)induction;
2. they are strongly related with \(\mu LK\): we can translate back and forth between \(\mu LK\) proofs and cyclic proofs;
3. they mirror closely the mathematical structure of Büchi automata and their computations: adequacy;
4. soundness and completeness of Büchi acceptance and inclusion;
\( \muLK = \muLK^\omega \subseteq \muLK^\infty \)

1. New logic \( \muLK^\omega \) and \( \muLK^\infty \) supporting mutual inductive and coinductive definitions with implicit (co)induction;
2. they are strongly related with \( \muLK \): we can translate back and forth between \( \muLK \) proofs and cyclic proofs;
3. they mirror closely the mathematical structure of Büchi automata and their computations: adequacy;
4. soundness and completeness of Büchi acceptance and inclusion;
5. first result for cut-elimination of infinite proofs.
Conclusion

The End

Thanks for listening!
References I


Nondeterministic

\[ L(A_1) = (0|1)^\omega \subseteq L(A_2) = (0|1)^\omega \]
Encoding

Encoding of $\mathcal{A} = (Q, \Sigma, \delta, Q_I, Q_F)$:

$$[\mathcal{A}] = \lambda w. \bigvee_{q \in Q_I} [q]^{\emptyset} w$$

$$[q]^{\gamma} = \begin{cases} q & \text{if } q \in \gamma \\ \mu \left( \lambda q. \lambda w. \exists w' \bigvee_{q' \in \delta(q, \alpha), \alpha \in \Sigma} w = \alpha \cdot w' \land [q']^{\gamma \cup \{q\}} w' \right) & \text{if } q \in Q_F \\ \nu \left( \lambda q. \lambda w. \exists w' \bigvee_{q' \in \delta(q, \alpha), \alpha \in \Sigma} w = \alpha \cdot w' \land [q']^{\gamma \cup \{q\}} w' \right) & \text{else} \end{cases}$$
Adequacy

- \( w \in L(\mathcal{A}) \iff \vdash \llbracket \mathcal{A} \rrbracket[w] \): The proof tries all the possible runs in parallel.
Adequacy

- $w \in L(\mathcal{A}) \iff \vdash \llbracket A \rrbracket [w]$: The proof tries all the possible runs in parallel.
  $w \in L(\mathcal{A}) \iff$ there is at least one accepted run $\iff$ there is at least one valid observation $\iff \vdash \llbracket A \rrbracket [w]$ is provable;
- There is a bijection between the runs and the observations of the proof $\vdash \llbracket A \rrbracket [w]$. 
\( \mathcal{L}(\mathcal{A}_1) \subseteq \mathcal{L}(\mathcal{A}_2) \Rightarrow \|A_1\|_x \vdash \|A_2\|_x: \)

- We prove the inclusion in \( \mu \text{LK}^\infty \):

\[ \|A_1\|_x \vdash \|A_2\|_x \Rightarrow \mathcal{L}(\mathcal{A}_1) \subseteq \mathcal{L}(\mathcal{A}_2): \]
\( \mathcal{L}(\mathcal{A}_1) \subseteq \mathcal{L}(\mathcal{A}_2) \Rightarrow [\! [A_1] \! ] x \vdash [\! [A_2] \! ] x: \)

- We prove the inclusion in \( \mu \text{LK}^\infty \):
  - on the right we test all the runs in \( \mathcal{A}_2 \) in parallel;

\[ [\! [A_1] \! ] x \vdash [\! [A_2] \! ] x \Rightarrow \mathcal{L}(\mathcal{A}_1) \subseteq \mathcal{L}(\mathcal{A}_2): \]
\[ L(A_1) \subseteq L(A_2) \Rightarrow \| A_1 \| x \vdash \| A_2 \| x: \]

- We prove the inclusion in \( \mu \text{LK}^\infty \):
  - on the right we test all the runs in \( A_2 \) in parallel;
  - on the left we branch at each disjunction: each infinite branch denotes a run in \( A_1 \);

\[ \| A_1 \| x \vdash \| A_2 \| x \Rightarrow L(A_1) \subseteq L(A_2): \]
$L(A_1) \subseteq L(A_2) \Rightarrow \lbrack A_1 \rbrack x \vdash \lbrack A_2 \rbrack x$:
- We prove the inclusion in $\mu LK^\omega$:
  - on the right we test all the runs in $A_2$ in parallel;
  - on the left we branch at each disjunction: each infinite branch denotes a run in $A_1$;
  - for each branch of the proof:
    - if the run on the left is valid, then the word is in $L(A_1)$ so is in $L(A_2)$.
    - else the observation of $T_{A_1} U \lbrack w \rbrack$ in this branch is valid.
    - then these proofs are regular so are in $\mu LK$;
    - then we can build a proof in $\mu LK$.

$\lbrack A_1 \rbrack x \vdash \lbrack A_2 \rbrack x \Rightarrow L(A_1) \subseteq L(A_2)$:
\[ \mathcal{L}(\mathcal{A}_1) \subseteq \mathcal{L}(\mathcal{A}_2) \Rightarrow \| A_1 \| x \vdash \| A_2 \| x : \]

- We prove the inclusion in \(\mu LK^\infty\):
  - on the right we test all the runs in \(\mathcal{A}_2\) in parallel;
  - on the left we branch at each disjunction: each infinite branch denotes a run in \(\mathcal{A}_1\);
  - for each branch of the proof:
    - if the run on the left is valid, then the word is in \(\mathcal{L}(\mathcal{A}_1)\) so is in \(\mathcal{L}(\mathcal{A}_2)\). Then one of the run on the right is valid;

\[ \| A_1 \| x \vdash \| A_2 \| x \Rightarrow \mathcal{L}(\mathcal{A}_1) \subseteq \mathcal{L}(\mathcal{A}_2) : \]
\( \mathcal{L}(\mathcal{A}_1) \subseteq \mathcal{L}(\mathcal{A}_2) \Rightarrow \| A_1 \| x \vdash \| A_2 \| x : \)

- We prove the inclusion in \( \mu LK^\infty : \)
  - on the right we test all the runs in \( \mathcal{A}_2 \) in parallel;
  - on the left we branch at each disjunction: each infinite branch denotes a run in \( \mathcal{A}_1 \);
  - for each branch of the proof:
    - if the run on the left is valid, then the word is in \( \mathcal{L}(\mathcal{A}_1) \) so is in \( \mathcal{L}(\mathcal{A}_2) \). Then one of the run on the right is valid;
    - else the observation of \( \| \mathcal{A}_1 \| \) in this branch is valid.

\[ \| A_1 \| x \vdash \| A_2 \| x \Rightarrow \mathcal{L}(\mathcal{A}_1) \subseteq \mathcal{L}(\mathcal{A}_2) : \]
\[ \mathcal{L}(\mathcal{A}_1) \subseteq \mathcal{L}(\mathcal{A}_2) \Rightarrow \sem{\mathcal{A}_1} x \vdash \sem{\mathcal{A}_2} x : \]

- We prove the inclusion in \( \mu \text{LK}^\infty \):
  - on the right we test all the runs in \( \mathcal{A}_2 \) in parallel;
  - on the left we branch at each disjunction: each infinite branch denotes a run in \( \mathcal{A}_1 \);
  - for each branch of the proof:
    - if the run on the left is valid, then the word is in \( \mathcal{L}(\mathcal{A}_1) \) so is in \( \mathcal{L}(\mathcal{A}_2) \). Then one of the run on the right is valid;
    - else the observation of \( \sem{\mathcal{A}_1} \) in this branch is valid.

- then these proofs are regular so are in \( \mu \text{LK}^\omega \);

\[ \sem{\mathcal{A}_1} x \vdash \sem{\mathcal{A}_2} x \Rightarrow \mathcal{L}(\mathcal{A}_1) \subseteq \mathcal{L}(\mathcal{A}_2) : \]
\( \mathcal{L}(A_1) \subseteq \mathcal{L}(A_2) \Rightarrow \|A_1\|_x \vdash \|A_2\|_x: \)

- We prove the inclusion in \( \mu \text{LK}^\infty \):
  - on the right we test all the runs in \( A_2 \) in parallel;
  - on the left we branch at each disjunction: each infinite branch denotes a run in \( A_1 \);
  - for each branch of the proof:
    - if the run on the left is valid, then the word is in \( \mathcal{L}(A_1) \) so is in \( \mathcal{L}(A_2) \). Then one of the run on the right is valid;
    - else the observation of \( \|A_1\| \) in this branch is valid.

- then these proofs are regular so are in \( \mu \text{LK}^\omega \);

- then we can build a proof in \( \mu \text{LK} \).

\( \|A_1\|_x \vdash \|A_2\|_x \Rightarrow \mathcal{L}(A_1) \subseteq \mathcal{L}(A_2): \)
\[ L(\mathcal{A}_1) \subseteq L(\mathcal{A}_2) \Rightarrow \llbracket A_1 \rrbracket x \vdash \llbracket A_2 \rrbracket x : \]
- We prove the inclusion in \( \mu LK^\infty \):
  - on the right we test all the runs in \( \mathcal{A}_2 \) in parallel;
  - on the left we branch at each disjunction: each infinite branch denotes a run in \( \mathcal{A}_1 \);
  - for each branch of the proof:
    - if the run on the left is valid, then the word is in \( L(\mathcal{A}_1) \) so is in \( L(\mathcal{A}_2) \). Then one of the run on the right is valid;
    - else the observation of \( \llbracket A_1 \rrbracket \) in this branch is valid.

- then these proofs are regular so are in \( \mu LK^\omega \);
- then we can build a proof in \( \mu LK \).

\[ \llbracket A_1 \rrbracket x \vdash \llbracket A_2 \rrbracket x \Rightarrow L(\mathcal{A}_1) \subseteq L(\mathcal{A}_2) : \]
- If we prove the inclusion in one of the logics we can prove it in \( \mu LK^\infty \);
\[ \mathcal{L}(A_1) \subseteq \mathcal{L}(A_2) \Rightarrow \models A_1 \vdash \models A_2 : \]

- We prove the inclusion in \( \mu\text{LK}^\infty \):
  - on the right we test all the runs in \( A_2 \) in parallel;
  - on the left we branch at each disjunction: each infinite branch
denotes a run in \( A_1 \);
  - for each branch of the proof:
    - if the run on the left is valid, then the word is in \( \mathcal{L}(A_1) \) so is in
      \( \mathcal{L}(A_2) \). Then one of the run on the right is valid;
    - else the observation of \( \models A_1 \) in this branch is valid.
  - then these proofs are regular so are in \( \mu\text{LK}^\omega \);
  - then we can build a proof in \( \mu\text{LK} \).

\[ \models A_1 \vdash \models A_2 \Rightarrow \mathcal{L}(A_1) \subseteq \mathcal{L}(A_2) : \]

- If we prove the inclusion in one of the logics we can prove it in
  \( \mu\text{LK}^\infty \);
- if \( w \in \mathcal{L}(A_1) \) then \( \Pi_1 : \vdash \models A_1 \models w \) and:

\[
\begin{array}{c}
\Pi_1 \\
\vdash \models A_1 \models w
\end{array} \quad \text{and} \quad
\begin{array}{c}
\Pi_2 \\
\models A_1 \models w \vdash \models A_2 \models w
\end{array}
\]

\[ \vdash \models A_2 \models w \quad \text{cut} \]
Results

$\mu LK \subseteq \mu LK^\omega \subseteq \mu LK^\infty$

Inclusions of the Logics

**Theorem 1**

$\mu LK \subseteq \mu LK^\omega \subseteq \mu LK^\infty$

$LK^\omega \subseteq \mu LK^\infty$: not the same language. We need a translation and a table of (co)-induction.
Inclusions of the Logics

Theorem 1

\[ \mu LK \subseteq \mu LK^\omega \subseteq \mu LK^\infty \]

- \( LK^\omega \subseteq \mu LK^\infty \): unfold the cycles infinitely often.

\[ \Psi \vdash \alpha \]

\[ \Psi \Gamma \vdash P \]

\[ \Pi \]

\[ \Gamma \vdash P : \alpha \]

\[ \Psi \vdash P \diamond \]

\[ \Pi \]

\[ \Gamma \vdash P \diamond \]
Inclusions of the Logics

**Theorem 1**

\[ \mu LK \subseteq \mu LK^\omega \subseteq \mu LK^\infty \]

- \( LK^\omega \subseteq \mu LK^\infty \): unfold the cycles infinitely often.

\[ \Psi \leftrightarrow \alpha \]

\[ \frac{\Psi \Gamma \vdash P}{\Gamma \vdash P : \alpha} \Rightarrow \quad \frac{\Psi \Gamma \vdash P \diamond}{\Gamma \vdash P \diamond} \]

- \( \mu LK \subseteq \mu LK^\omega \): not the same language. We need a translation and a table of (co)-induction.
Table of (co)-induction:

- $Q = \{ \hat{\varepsilon} B \mid B \text{ closed operator of } \mu LK, \varepsilon \in \{\mu; \nu\}\}; \quad \varepsilon(\hat{\varepsilon} B) = \varepsilon;$
Table of (co)-induction:

- $Q = \{ \varepsilon B \mid B \text{ closed operator of } \muLK, \varepsilon \in \{\mu, \nu\} \}; \quad \varepsilon(\varepsilon B) = \varepsilon$;

\[
\langle \_ \rangle : \muLK \text{ formula} \rightarrow \muLK^\omega \text{ formula}
\]

\[
\langle P \Box Q \rangle = \langle P \rangle \Box \langle Q \rangle \quad \Box \in \{\land, \lor, \Rightarrow\}
\]

\[
\langle \otimes x B \rangle = \otimes x \langle B \rangle \quad \otimes \in \{\forall, \exists\}
\]

\[
\langle \varepsilon B \rangle = \varepsilon B \quad \varepsilon \in \{\mu, \nu\}
\]

\[
\langle a \rangle = a
\]
Results

$\mu LK \subseteq \mu LK^\omega \subseteq \mu LK^\infty$

Table of (co)-induction:

- $Q = \{ \widehat{\varepsilon B} \mid B \text{ closed operator of } \mu LK, \varepsilon \in \{\mu; \nu\}\}$; $\varepsilon(\widehat{\varepsilon B}) = \varepsilon$;

\[
\langle \_ \rangle : \mu LK \text{ formula } \rightarrow \mu LK^\omega \text{ formula}
\]

\[
\langle P \Box Q \rangle = \langle P \rangle \Box \langle Q \rangle \quad \Box \in \{\land; \lor; \Rightarrow\}
\]

\[
\langle \otimes x B \rangle = \otimes x \langle B \rangle \quad \otimes \in \{\forall; \exists\}
\]

\[
\langle \varepsilon B \rangle = \widehat{\varepsilon B}
\]

\[
\langle a \rangle = a
\]

- $\widehat{\varepsilon B} \supseteq \langle B \varepsilon B \rangle$;
Table of (co)-induction:

- $Q = \{ \widehat{\varepsilon B} \mid B \text{ closed operator of } \mu \text{LK}, \varepsilon \in \{\mu; \nu\} \}; \quad \varepsilon(\widehat{\varepsilon B}) = \varepsilon$;

$$
\langle \_ \rangle: \mu \text{LK formula } \rightarrow \mu \text{LK}^\omega \text{ formula}
$$

- $\langle P \square Q \rangle = \langle P \rangle \square \langle Q \rangle \quad \square \in \{\wedge; \vee; \Rightarrow\}$
- $\langle \otimes x B \rangle = \otimes x \langle B \rangle \quad \otimes \in \{\forall; \exists\}$
- $\langle \varepsilon B \rangle = \widehat{\varepsilon B}$
- $\langle a \rangle = a$

- $\widehat{\varepsilon B} \geq \langle B \varepsilon B \rangle$;
- $\widehat{\varepsilon B} < \widehat{\varepsilon' B'} \iff \varepsilon' B' \text{ sub-formula of } B$
\[\mu LK \subseteq \mu LK^\omega\]

Lemma: Functoriality in $\mu LK^\omega$

If $B$ is monotonic (i.e. the $p_i$ appears only in positive positions in $B$) then for all predicates $P_1, P_2, \ldots P_n$ this rule is admissible in $\mu LK^\omega$:

Let $B$ a predicate operator: $B = \lambda p. \lambda x. A$ and $P$ and $Q$ some predicates then this rule is admissible in $\mu LK^\omega$:

\[
\frac{\langle P \rangle \ x \vdash \langle Q \rangle \ x}{\langle B \ P \rangle \ t \vdash \langle B \ Q \rangle \ t} \quad \text{functo}
\]

and all the observations involve names $n$ such that for all names $m$ appearing in $\langle P \rangle$ or $\langle Q \rangle$, $n < m$. 
Theorem 1

\( \Gamma \vdash_{\mu LK} \Delta \Rightarrow \langle \Gamma \rangle \vdash_{\mu LK}^{\omega} \langle \Delta \rangle \)

By induction on the size of the proof then case analysis on the first rule.

By induction on the size of the proof then case analysis on the first rule.

\[
\frac{\Pi_1}{\Gamma \vdash \Delta, S \, t} \quad \frac{\Pi_2}{S \, x \vdash BS \, x} \quad \frac{\nu R}{\Gamma \vdash \Delta, \nu B \, t}
\]

\[
\downarrow
\]

By induction on the size of the proof then case analysis on the first rule.

\[
\frac{\Pi_2^*}{\langle S \rangle \, x \vdash \langle BS \rangle \, x} \quad \frac{\langle S \rangle \, x, \langle S \rangle \, x \vdash \nu B \, x}{WL} \quad \frac{\langle S \rangle \, x, \langle BS \rangle \, x \vdash \langle B(\nu B) \rangle \, x}{functo} \quad \frac{\langle S \rangle \, x \vdash \langle B(\nu B) \rangle \, x}{cut} \quad \frac{\nu R}{\langle S \rangle \, x \vdash \widehat{\nu B} \, x : \alpha}
\]

By induction on the size of the proof then case analysis on the first rule.

\[
\frac{\Pi_1^*}{\langle \Gamma \rangle \vdash \langle \Delta \rangle, \langle S \rangle \, t} \quad \frac{\langle \Gamma \rangle \vdash \langle \Delta \rangle, \widehat{\nu B} \, t}{cut, \ cut, \ \forall R, \Rightarrow \ R, \ ...}
\]
Theorem 2

\[ \mu LK^\omega \subseteq \mu LK \]

Soon a complete proof.
Cut Elimination

Proof of normalisation of $\mu LK^\infty$. 

We must show:

1. Normalisation: the reduction rules provide a limit proof;
2. Validity: the limit proof is also valid.

We focus on a sub-logic containing only (co)-inductive formula: $\mu L_0$.

{given a formula: unique infinite observation.

Exploration of the reduction: The sub-part of the proof which is explored by the reduction.
Cut Elimination

Proof of normalisation of $\mu LK^\infty$. We must show:

1. **normalisation**: the reduction rules provides a limit proof;

   $$d(\Pi, \Pi') = \frac{1}{1 + \text{minimum depth of two different nodes}}$$

2. **validity**: the limit proof is also valid.
Cut Elimination

Proof of normalisation of $\mu LK^\infty$. We must show:

1. **normalisation**: the reduction rules provide a limit proof;

$$d(\Pi, \Pi') = \frac{1}{1 + \text{minimum depth of two different nodes}}$$

2. **validity**: the limit proof is also valid.

We focus on a sub-logic containing only (co)-inductive formula: $\mu L_0$. 
Cut Elimination

Proof of normalisation of $\mu LK^\infty$. We must show:

1. **normalisation**: the reduction rules provide a limit proof;

$$d(\Pi, \Pi') = \frac{1}{1 + \text{minimum depth of two different nodes}}$$

2. **validity**: the limit proof is also valid.

We focus on a sub-logic containing only (co)-inductive formula:

$\mu L_0$.

$\leadsto$ given a formula: unique infinite observation.
Cut Elimination

Proof of normalisation of $\mu LK^\infty$. We must show:

1. **normalisation**: the reduction rules provides a limit proof;

$$d(\Pi, \Pi') = \frac{1}{1 + \text{minimum depth of two different nodes}}$$

2. **validity**: the limit proof is also valid.

We focus on a sub-logic containing only (co)-inductive formula: $\mu L_0$.

$\rightsquigarrow$ given a formula: unique infinite observation.

**Exploration of the reduction**

The sub-part of the proof which is explored by the reduction.
Strategy of reduction

Always reduce the first cut rule which is no followed by another cut rule.
Strategy of reduction

Always reduce the first cut rule which is no followed by another cut rule.

Lemma 1: Exploration

With this strategy, the exploration is connex.
Strategy of reduction
Always reduce the first cut rule which is no followed by another cut rule.

Lemma 1: Exploration
With this strategy, the exploration is connex.

Lemma 2: Dual observations
Two dual observations can not be both valid.
Strategy of reduction

Always reduce the first cut rule which is not followed by another cut rule.

Lemma 1: Exploration

With this strategy, the exploration is connex.

Lemma 2: Dual observations

Two dual observations cannot be both valid.

Lemma 3

For a cut rule: $\frac{\Pi_1 \Pi_2}{s} \text{cut}$. If there is an infinite observation of the cut formula in $\Pi_i$ contained in the exploration, then there is a dual observation in $\Pi_{1-i}$ in the exploration.
Lemma 4

There exists an infinite branch in the exploration which has a valid observation of a formula in the root.
Lemma 4

There exists an infinite branch in the exploration which has a valid observation of a formula in the root.

Lemma 5

If there exists an observation from the root in the exploration, then the reduction produces at least the sequents of it.
Lemma 4

There exists an infinite branch in the exploration which has a valid observation of a formula in the root.

Lemma 5

If there exists an observation from the root in the exploration, then the reduction produces at least the sequents of it.

Lemma 4 + Lemma 5 $\sim$ Normalisation + Validity!
Results

- Infinite proofs: cut-elimination, regular proofs $= \mu\text{LK}^\omega$

$$\mu\text{LK} = \mu\text{LK}^\omega \subseteq \mu\text{LK}^\infty$$
Results

- Infinite proofs: cut-elimination, regular proofs $= \mu LK^\omega$
- $\mu LK$: cut-elimination; as expressive as $\mu LK^\omega$

$\mu LK = \mu LK^\omega \subseteq \mu LK^\infty$
Results

- Infinite proofs: cut-elimination, regular proofs $= \mu LK^\omega$
- $\mu LK$: cut-elimination; as expressive as $\mu LK^\omega$

$\mu LK = \mu LK^\omega \subseteq \mu LK^\infty$

- Cyclic proofs: consistent, as expressive as $\mu LK$
Outline

6 Results

7 Mental Repository
Encoding

Encoding from the Büchi automata to formulas of a logic so as to reason over the automata within the logic.

We must trust the encoding (and the logic) for working within the logic instead of manipulating automata directly.