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

Lucca Hirschi

August 29, 2012

Lucca Hirschi

directed by

David Baelde

ENS Lyon

 directed by

ITU of Copenhagen
The purpose in one sentence

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

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; hosting Büchi Automata.
Case of Finite Automata

A complete solution is given in the thesis [? ].
Case of Finite Automata

A complete solution is given in the thesis [? ]. The logic is $\mu$LJ: an intuitionistic sequent calculus with explicit (co)-induction rules.
Case of Finite Automata

A complete solution is given in the thesis [? ]. The logic is $\mu$LJ: an intuitionistic sequent calculus with explicit (co)-induction rules.

$$\text{Nat} = \mu (\lambda N. \lambda n. \ n = 0 \lor (\exists n' \ n = s(n') \land N n'))$$
Case of Finite Automata

A complete solution is given in the thesis [? ]. The logic is $\mu LJ$: an intuitionistic sequent calculus with explicit (co)-induction rules.

$$\text{Nat} = \mu (\lambda N. \lambda n. n = 0 \lor (\exists n' n = s(n') \land N n'))$$

$\llbracket A \rrbracket = [q_0]$

$[q_0] = \lambda w. w = 1 : w' \land [q_1] w'$

$[q_1] = \mu (\lambda q_1. \lambda w. w = \epsilon \lor w = 0 : w' \land q_1 w')$
Case of Finite Automata

\[
\begin{align*}
\llbracket \mathcal{A} \rrbracket &= [q_0] \\
[q_0] &= \lambda w. w = 1 : w' \land [q_1] w' \\
[q_1] &= \mu (\lambda q_1. \lambda w. w = \epsilon \lor w = 0 : w' \land q_1 w')
\end{align*}
\]
\[ \mathcal{A} \] = \{ q_0 \}
\[ q_0 \] = \lambda w. \ w = 1: w' \land [q_1] w'
\[ q_1 \] = \mu (\lambda q_1. \lambda w. \ w = \epsilon \lor w = 0: w' \land q_1 w')
Büchi Automata

Used to describe infinite objects by finite automata.
Büchi Automata

Used to describe infinite objects by finite 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.

The same structure as finite automata but not the same acceptance condition.
Büchi Automata

<table>
<thead>
<tr>
<th>Definition (Acceptance condition)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A run $\alpha$ on a word is <strong>accepted</strong> by an automaton $\iff$ $\alpha$ visits a final state infinitely often;</td>
</tr>
<tr>
<td>A word is <strong>recognized</strong> by an automaton $\iff$ there exists an accepting run on it.</td>
</tr>
</tbody>
</table>
Introduction

Definition: Büchi Automata

**Büchi Automata**

Definition (Acceptance condition)

- A run $\alpha$ on a word is **accepted** 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.

$\rightarrow$ Büchi Automata recognize only infinite words.
An Example of a Büchi Automaton

The Büchi automaton $\mathcal{A}$.

$$\mathcal{L}(\mathcal{A}) = 0^+1(10^+1)^\omega$$
Requirements

- **Adequacy:** \( w \in \mathcal{L}(A) \iff \vdash \llbracket A \rrbracket[w] \)
Requirements

- **Adequacy:** \( w \in \mathcal{L}(A) \iff \vdash \llbracket A \rrbracket[w] \) and a bijection between the runs on a word \( w \) and the normal proofs \( \vdash \llbracket A \rrbracket[w] \);
Requirements

- **Adequacy**: \( w \in L(A) \iff \vdash \llbracket A \rrbracket [w] \) and a bijection between the runs on a word \( w \) and the normal proofs \( \vdash \llbracket A \rrbracket [w] \);

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

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

- **Adequacy:** \( w \in \mathcal{L}(A) \iff \vdash \llbracket A \rrbracket[w] \) and a bijection between the runs on a word \( w \) and the normal proofs \( \vdash \llbracket A \rrbracket[w] \);

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

- **Adequacy:** $w \in \mathcal{L}(\mathcal{A}) \iff \vdash [\mathcal{A}][w]$ and a bijection between the runs on a word $w$ and the normal proofs $\vdash [\mathcal{A}][w]$;

- **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 $\sim$ inclusion and a certificate;

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

What kind of logics do we need?
Induction and Co-induction Interleaved

**Finite Automata**

- loop $\leadsto$ inductive formula
- final state $\leadsto$ trivial formula
Induction and Co-induction Interleaved

Finite Automata

- Loop \(\sim\) inductive formula
- Final state \(\sim\) trivial formula

Büchi Automata

- Loop which visits a final state \(\sim\) co-inductive formula
Induction and Co-induction Interleaved

**Finite Automata**

- loop $\rightsquigarrow$ inductive formula
- final state $\rightsquigarrow$ trivial formula

**Büchi Automata**

- loop which visits a final state $\rightsquigarrow$ co-inductive formula
- loop which does not visit a final state $\rightsquigarrow$ inductive formula
Induction and Co-induction Interleaved

**Finite Automata**

- loop $\leadsto$ inductive formula
- final state $\leadsto$ trivial formula

**Büchi Automata**

- loop which visits a final state $\leadsto$ co-inductive formula
- loop which does not visit a final state $\leadsto$ inductive formula

Mixing and interleaving inductive and co-inductive formulas is hard.
Adequacy

There are uncountably many runs. Example:
Adequacy

There are uncountably many runs. Example:

\[
\begin{array}{c}
q \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
q' \\
\end{array}
\]

Adequacy \(\leadsto\) uncountably many proofs.
Adequacy

There are uncountably many runs. Example:

\[ \xrightarrow{0} q \xrightarrow{0} q' \xrightarrow{0} q \xrightarrow{0} q' \]

Adequacy \(\leadsto\) uncountably many proofs.

\(\leadsto\) We must consider infinite proofs.
Synopsis

- Infinite proofs: satisfies adequacy, impractical

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

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

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

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

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

- Cyclic proofs
Internship at ITU of Copenhagen: From Büchi Automata to Cyclic and Infinite Proofs

Outline

1. Introduction
2. Logics
3. Results
4. Büchi Automata within the Logics
5. Conclusion
Outline

1. Introduction
2. Logics
3. Results
4. Büchi Automata within the Logics
5. Conclusion
**Definition (Formula of $\mu$LK)**

$P ::= \top | \bot$

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

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

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

\[ P ::= \top \mid \bot \mid \exists x. \ P \mid \forall x. \ P \mid P \land P \mid P \lor P \mid P \Rightarrow P \mid s = t \mid p \mid \mu(\lambda p. \lambda x_1 \ldots \lambda x_n. \ P) \ t_1 \ldots t_n \mid \nu(\lambda p. \lambda x_1 \ldots \lambda x_n. \ P) \ t_1 \ldots t_n \]

- \( x \in \mathcal{V} \)
- \( t, s \) some terms
- \( p \in \mathcal{V}_f \)
- \( p \in \mathcal{V}_f, \ t_i \) a term
**Definition (Formula of $\mu LK$)**

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

\[
S = \nu B_{\text{stream}} = \nu (\lambda p_s. \lambda w. \exists w' \forall n \ w = n : w' \land N \ n \land p_s \ w')
\]

\[
N = \mu B_{\text{nat}} = \mu (\lambda p_n. \lambda x. x = 0 \lor (\exists y \ x = s(y) \land p_n \ y))
\]
Rules of $\mu LK$

Sequent calculus:

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

Sequent calculus:

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

$+$ explicit (co)-induction:

$$
\frac{\Gamma \vdash B(\mu B) \mathbf{t}}{\Gamma \vdash \mu B \mathbf{t}} \mu R \quad \frac{\Gamma, S \mathbf{t} \vdash P \quad BS \mathbf{x} \vdash S \mathbf{x}}{\Gamma, \mu B \mathbf{t} \vdash P} \mu L
$$

$$
\frac{\Gamma \vdash St \quad St \vdash BSst}{\Gamma \vdash \nu B \mathbf{t}} \nu R \quad \frac{\Gamma, B(\nu B) \mathbf{t} \vdash P}{\Gamma, \nu B \mathbf{t} \vdash P} \nu L
$$
\[
\begin{array}{c}
\Gamma \vdash B(\mu B) \quad \frac{\mu R}{\Gamma \vdash \mu B} \\
\Gamma, S \vdash P \quad \frac{BS x \vdash S x}{\mu L}
\end{array}
\]

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

$\Gamma \vdash \mu B \quad \mu \mathcal{R}$

$\Gamma, S \vdash P \quad BS \quad \Psi_0 \quad \Psi_n$

$\Psi_0$

$\Psi_n$

$x = 0 \lor \exists y \quad x = s(y) \land S \quad y \vdash S \quad x$

$\land \mathcal{L}, (\exists \mathcal{L}), = \mathcal{L}$

$\mu \mathcal{L}$
\( \mu LK \) enjoys cut-elimination [? ].
Explicit (co)-induction rules \(\sim\) replaced by some cycles.
Explicit (co)-induction rules $\leadsto$ replaced by some cycles.

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

\[
\begin{align*}
\frac{t = 0 \vdash \mu B_{\text{nat}} t}{t = 0 \vdash \mu B_{\text{nat}} t} & \quad \frac{t = s(s(t')) \land \mu B_{\text{even}} t' \vdash \mu B_{\text{nat}} t}{t = s(s(t')) \land \mu B_{\text{even}} t' \vdash \mu B_{\text{nat}} t} & \quad \frac{\mu B_{\text{even}} t' \vdash \mu B_{\text{nat}} (s(s(t')))}{\mu B_{\text{even}} t' \vdash \mu B_{\text{nat}} (s(s(t')))}
\end{align*}
\]

$\mu L'$
Explicit (co)-induction rules $\leadsto$ replaced by some cycles.

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

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

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

\[
\mu L'
\]
Explicit (co)-induction rules $\leadsto$ replaced by some cycles.

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

\begin{align*}
\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')))) \\
\mu B_{\text{even}} t & \vdash \mu B_{\text{nat}} t \\
\mu B_{\text{even}} t' & \vdash \mu B_{\text{nat}} t : \alpha
\end{align*}
Explicit (co)-induction rules $\leadsto$ replaced by some cycles.

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

\[
\begin{array}{c}
\mu B_{\text{even}} \ t'' \vdash \mu B_{\text{nat}} \ t'' : \Diamond \\
\mu B_{\text{even}} \ t' \vdash \mu B_{\text{nat}} \ t' : \Diamond \\
\mu B_{\text{even}} \ t \vdash \mu B_{\text{nat}} \ t : \Diamond
\end{array}
\]

$$
\mu L' \ \mu L'
$$
Guard Condition

Litterature:

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

Litterature:

1. Brotherstone: No co-inductive formula; “infinite descent”; [? ]
2. Santocanale: No cut rule; inductive and co-inductive formula; [? ]
Litterature:

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

First bug

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

\[
\begin{array}{c}
\alpha \\
\nu L \\
\mu L \\
\text{cut}
\end{array}
\begin{array}{c}
\mu R \\
\nu R \\
\nu L \\
\text{cut}
\end{array}
\begin{array}{c}
\tau \\
\alpha \\
\mu R
\end{array}
\begin{array}{c}
P \vdash \bot \\
Q \vdash \bot \\
P \vdash \bot \\
Q \vdash \bot \\
P \vdash \bot \\
Q \vdash \bot \\
P \vdash \bot \\
\vdash \bot
\end{array}
\]
Guard Condition - interleaved (co)-inductive formulas

First bug

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

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

\[
\mu R
\]

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

\[
\nu R
\]

\[
\begin{array}{c}
\vdash P
\end{array}
\]

\[
\nu L
\]

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

\[
\mu L
\]

\[
\vdash \perp
\]

\[
\text{cut}
\]

Fix

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

\[P > Q\]
Guard Condition - interleaved (co)-inductive formulas

First bug

\[ P = \mu(\lambda p. \nu(\lambda q. p)) \]
\[ Q = \nu(\lambda q. P) \]

\[ \vdash P \]
\[ \vdash Q \]
\[ \vdash \alpha \]
\[ \vdash Q \vdash \mu R \]
\[ \vdash P \vdash \nu R \]
\[ \vdash Q : \alpha \vdash \nu R \]
\[ \vdash P : \mu R \]
\[ \vdash \tau \]
\[ \vdash Q : \tau \vdash \nu L \]
\[ \vdash P : \tau \vdash \mu L \]
\[ \vdash \tau \]

Fix

\[ P =_\mu Q \]
\[ Q =_\nu P \]

\[ \vdash Q \]
\[ \vdash P \]
\[ \vdash \mu R_P \]
\[ \vdash Q : \alpha \vdash \nu R_Q \]
\[ \vdash P : \mu R_P \]
\[ \vdash \tau \]
\[ \vdash Q : \tau \vdash \nu L_Q \]
\[ \vdash P : \tau \vdash \mu L_P \]
\[ \vdash \tau \]
Guard Condition - interleaved (co)-inductive formulas

**First bug**

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

\[
\begin{align*}
\varphi \alpha & \vdash Q \\
\vdash P & \vdash \mu R \\
\vdash Q : \alpha & \vdash \nu R \\
\vdash P & \vdash \nu L \\
\vdash \perp & \vdash \mu \perp \\
\vdash \perp & \vdash \nu \perp \\
\vdash \perp : \tau & \vdash \mu L \ \
\text{cut}
\end{align*}
\]

**Fix**

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

\[
\begin{align*}
\varphi \alpha & \vdash Q \\
\vdash P & \vdash \mu R_P \\
\vdash Q : \alpha & \vdash \nu R_Q \\
\vdash P & \vdash \nu L_Q \\
\vdash \perp & \vdash \mu \perp \\
\vdash \perp & \vdash \nu \perp \\
\vdash \perp : \tau & \vdash \mu L_P \ \
\text{cut}
\end{align*}
\]
"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?
“Definition”: table of (co)-induction

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

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

Second bug

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

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

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

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

**Second bug**

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

\[
\begin{array}{c}
\frac{B_{\text{nat}} \text{Nat } t \vdash B_{\text{nat}} \text{Nat } t}{B_{\text{nat}} \text{Nat } t \vdash \text{Nat } t} \quad \text{Ax} \\
\frac{\mu R}{\mu L} \quad \neg \neg \\Rightarrow \alpha \\
\frac{\text{cut}}{\text{Nat } t \vdash \bot} \\
\frac{B_{\text{nat}} \text{Nat } t \vdash \bot}{\text{Nat } t \vdash \bot} \quad : \alpha \\
\end{array}
\]
\[
\begin{align*}
&s_5 : \text{Even } t' \vdash \text{Nat } t' \\
&s_4 : \text{Even } t' \vdash \text{Nat } (s(t')) \\
&s_3 : \text{Even } t' \vdash \text{Nat } (s(s(t')))) \\
&\quad t = 0 \vdash \text{Nat } t \\
&s_2 : t = s(s(t')) \land \text{Even } t' \vdash \text{Nat } t \\
&s_1 : (A =) \text{Even } t \vdash \text{Nat } t : \alpha \\
\end{align*}
\]
Logics

Guard Condition

\[
\begin{align*}
\alpha & \quad \leftrightarrow \\
\alpha & \quad \rightarrow \\
\Rightarrow & \quad \text{Nat } t' \quad \vdash \quad \text{Even } t' \\
\Rightarrow & \quad \text{Nat } t \quad \vdash \quad \text{Even } t \\
\Rightarrow & \quad \text{Nat } s(t') \quad \vdash \quad \text{Even } t' \\
\Rightarrow & \quad \text{Nat } s(s(t')) \quad \vdash \quad \text{Even } t \\
\Rightarrow & \quad \text{Nat } s(s(s(t'))) \quad \vdash \quad \text{Even } t'
\end{align*}
\]

\[O_A(\alpha) = (\mu L, \text{Even})\]
The trace of $A \in s_0$ in the cycle $s_0, s_1, \ldots, s_n$ is a serie of formulas $A_0, A_1, \ldots, A_n$ 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 cycle is the serie of $(r, A)$ where $r$ is a (co)-inductive rules applied to $A$ appearing in the trace.
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 \( \epsilon(\max(r, n) \in o \{n\}) = \mu R \);

cooprime cycle: there is an observation \( o \) on the right such that \( \epsilon(\max(r, n) \in o \{n\}) = \nu \).
Internship at ITU of Copenhagen: From Büchi Automata to Cyclic and Infinite Proofs

Logics

Guard Condition

Definition: Refinement of Guard Condition

A proof is valid ⇐⇒ each cycle is either inductive or co-inductive.

Inductive cycle: there is an observation o on the left such that

\[ \epsilon(\max(r, n) \in o \{n\}) = \mu R \]

Co-inductive cycle: there is an observation o on the right such that

\[ \epsilon(\max(r, n) \in o \{n\}) = \nu L \]

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

\[ \frac{B_{nat} \text{ Nat } t \vdash \bot \quad \mu L}{\text{ Nat } t \vdash \bot \quad : \alpha} \]
A proof is valid \( \iff \) each cycle is either inductive or co-inductive.

- **inductive cycle**: there is an observation \( \sigma \) on the left such that \( \epsilon\left(\max_{(r,n)\in\sigma}\{n\}\right) = \mu \);
\[
\frac{B_{\text{nat}} \; \text{Nat} \; t \vdash B_{\text{nat}} \; \text{Nat} \; t}{B_{\text{nat}} \; \text{Nat} \; t \vdash \text{Nat} \; t} \quad \text{Ax}
\]
\[
\frac{\mu R}{\text{Nat} \; t \vdash \text{Nat} \; t} \quad \text{⇓} \; \alpha
\]
\[
\frac{B_{\text{nat}} \; \text{Nat} \; t \vdash \bot}{\text{Nat} \; t \vdash \bot} \quad \text{cut}
\]
\[
\frac{B_{\text{nat}} \; \text{Nat} \; t \vdash \bot}{\text{Nat} \; t \vdash \bot} : \alpha \quad \mu L
\]

"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 \(\epsilon\left(\max_{(r,n)\in o} \{n\}\right) = \nu\).
“Definition”: Guard Condition for $\mu LK^\omega$

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 $\epsilon \left( \max_{(r,n) \in o} \{ n \} \right) = \nu$.

“Definition”: Guard Condition for $\mu LK^\infty$

A proof is valid $\iff$ each infinite branch 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 $\epsilon \left( \max_{(r,n) \in \text{Inf}(o)} \{ n \} \right) = \nu$. 
Outline

1. Introduction
2. Logics
3. Results
4. Büchi Automata within the Logics
5. Conclusion
Results

\[ \mu \text{LK} \subseteq \mu \text{LK}^\omega \subseteq \mu \text{LK}^\infty \]

Inclusions of the Logics

**Theorem 1**

\[ \mu \text{LK} \subseteq \mu \text{LK}^\omega \subseteq \mu \text{LK}^\infty \]

\[ \text{LK}^\omega \subseteq \mu \text{LK}^\infty : \]
Theorem 1

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

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

\[ \Psi \overset{\alpha}{\rightarrow} \Gamma \vdash P \]

\[ \Psi \vdash \Gamma \vdash P \diamond \]

\[ \Psi \vdash \Gamma \vdash P \diamond \]

\[ \Psi \vdash \Gamma \vdash P \diamond \]

\[ \Psi \vdash \Gamma \vdash P \diamond \]

\[ \Psi \vdash \Gamma \vdash P \diamond \]

\[ \Psi \vdash \Gamma \vdash P \diamond \]

\[ \Psi \vdash \Gamma \vdash P \diamond \]

\[ \Psi \vdash \Gamma \vdash P \diamond \]

\[ \Psi \vdash \Gamma \vdash P \diamond \]

\[ \Psi \vdash \Gamma \vdash P \diamond \]
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.

\[ \begin{array}{c}
\therefore \alpha \\
\Psi \Gamma \vdash P \\
\Pi \\
\Gamma \vdash P : \alpha \\
\Psi \Gamma \vdash P \Diamond \\
\Pi \\
\Gamma \vdash P \Diamond
\end{array} \]

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

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

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

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

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

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

\[ \langle \varepsilon B \rangle = \varepsilon B \]

\[ \langle a \rangle = a \]
Table of (co)-induction:

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

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

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

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

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

\[
\langle a \rangle = a
\]

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

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

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

- $\langle P \square Q \rangle = \langle P \rangle \square \langle Q \rangle$, $\square \in \{\wedge; \vee; \Rightarrow\}$
- $\langle \odot x B \rangle = \odot x \langle B \rangle$, $\odot \in \{\forall; \exists\}$
- $\langle \epsilon B \rangle = \widehat{\epsilon B}$, $\epsilon \in \{\mu; \nu\}$
- $\langle a \rangle = a$

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

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

$$
\frac{
\langle P \rangle \mathbf{x} \vdash \langle Q \rangle \mathbf{x}
}{
\langle B \ P \rangle \mathbf{t} \vdash \langle B \ Q \rangle \mathbf{t}
} \text{ functo}
$$

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

\[ \Gamma \vdash_{\muLK} \Delta \implies \langle \Gamma \rangle \vdash_{\muLK}^{\omega} \langle \Delta \rangle \]

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

\[ \Pi_1 \quad \Pi_2 \]
\[ \frac{\Gamma \vdash \Delta, S \ t}{\Gamma \vdash \Delta, \nuB \ t} \quad \frac{S x \vdash BS x}{\nuR} \]

\[ \downarrow \]

\[ \Pi_2^{*} \quad \Pi_1^{*} \]
\[ \frac{\langle S \rangle x \vdash \nuB x}{\langle S \rangle x, \langle S \rangle x \vdash \nuB x} \quad \frac{\langle S \rangle x, \langle BS \rangle x \vdash \langle B(\nuB) \rangle x}{\langle S \rangle x \vdash \langle B(\nuB) \rangle x} \quad \text{WL} \]
\[ \frac{\langle S \rangle x \vdash \nuB x}{\langle S \rangle x \vdash \nuB x : \alpha} \quad \nuR \]

\[ \frac{\langle S \rangle x \vdash \nuB x : \alpha}{\langle S \rangle x, \langle BS \rangle x \vdash \langle B(\nuB) \rangle x} \quad \text{cut} \]

\[ \frac{\langle S \rangle x \vdash \nuB x : \alpha}{\langle S \rangle x \vdash \nuB x : \alpha} \quad \text{cut, cut, } \forall R, \implies R, \ldots \]
Theorem 2

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

Soon a complete proof.
Cut Elimination

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

Proof of normalisation of $\mu\text{LK}^\infty$. 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 $\mu L K^\infty$. 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: $\mu L_0$. 
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: $\mu L_0$.

$\leadsto$ 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: $\mu L_0$.

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

**Exploration of the reduction**

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

Cut Elimination

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 not 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

\[
\frac{\Pi_1 \quad \Pi_2}{s} \quad \text{cut}
\]

For a cut rule: \( \frac{\Pi_1 \quad \Pi_2}{s} \quad \text{cut} \). 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 $\implies$ 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 $= \mu\text{LK}^\omega$
- $\mu\text{LK}$: cut-elimination; as expressive as $\mu\text{LK}^\omega$

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

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

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

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

1. Introduction
2. Logics
3. Results
4. Büchi Automata within the Logics
5. Conclusion
Results

- **Encoding** from Büchi Automata to $\mu LK$, $\mu LK^\omega$ or $\mu LK^\infty$;
Results

- Encoding from Büchi Automata to $\mu LK, \mu LK^\omega$ or $\mu LK^\infty$;
- Adequacy in $\mu LK^\infty$;
Results

- **Encoding** from Büchi Automata to $\mu$LK, $\mu$LK$^\omega$ or $\mu$LK$^\infty$;
- **Adequacy** in $\mu$LK$^\infty$;
- **Completeness and correctness of inclusion** for $\mu$LK$^\infty$, $\mu$LK$^\omega$ and $\mu$LK.
Conclusion

What’s new?

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

- Cyclic proofs with induction, co-induction and cut-rules with clearly expressed guard condition which is consistent;
Conclusion

What’s new?

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

- **Cyclic** proofs with induction, co-induction and cut-rules with clearly expressed guard condition which is consistent;
- **Infinite** proofs with induction, co-induction and cut-rules with clearly expressed guard condition which satisfies cut-elimination;
Conclusion

What’s new?

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

- Cyclic proofs with induction, co-induction and cut-rules with clearly expressed guard condition which is consistent;
- Infinite proofs with induction, co-induction and cut-rules with clearly expressed guard condition which satisfies cut-elimination;
- A new approach to normalisation proofs dealing with infinite proofs;

These new logics are strongly related to the common logic \( \mu LK \);

These new logics can well host the Büchi Automata.
Conclusion

What’s new?

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

- **Cyclic** proofs with induction, co-induction and cut-rules with clearly expressed guard condition which is consistent;
- **Infinite** proofs with induction, co-induction and cut-rules with clearly expressed guard condition which satisfies cut-elimination;
- A new approach to normalisation proofs dealing with infinite proofs;
- These new logics are strongly related to the common logic $\mu LK$;
Conclusion

What’s new?

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

- **Cyclic** proofs with induction, co-induction and cut-rules with clearly expressed **guard condition** which is **consistent**;
- **Infinite** proofs with induction, co-induction and cut-rules with clearly expressed **guard condition** which satisfies **cut-elimination**;
- A new approach to **normalisation proofs** dealing with **infinite proofs**;
- These new logics are **strongly related** to the common logic \( \mu LK \);
- These new logics can well **host** the Büchi Automata.
The End

Thanks for listening!
References I


[] David Baelde. Least and greatest fixed points in linear logic. 13(1), January 2012. ACM Transactions on Computational Logic.


Conclusion

What does it remain?

- General proof of cut-elimination;
Conclusion

What does it remain?

- General proof of cut-elimination;
- Conjecture: $\mu\text{LK}^\omega \subseteq \mu\text{LK}$;
Conclusion

What does it remain?

- General proof of cut-elimination;
- Conjecture: \( \mu LK^\omega \subseteq \mu LK \);
- Define an algorithm which builds a cyclic proof of inclusion when it is possible.
Outline

6 Problems Caused by the Büchi Automata

7 Büchi Automata within the Logics

8 Mental Repository
Non-deterministic

\[ \mathcal{L}(A_1) = (0|1)^\omega \subseteq \mathcal{L}(A_2) = (0|1)^\omega \]
Outline

6 Problems Caused by the Büchi Automata

7 Büchi Automata within the Logics

8 Mental Repository
Internship at ITU of Copenhagen: From Büchi Automata to Cyclic and Infinite Proofs

Büchi Automata within the Logics

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

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

- $w \in L(\mathcal{A}) \iff \vdash \llbracket \mathcal{A} \rrbracket [w]$: 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 $\iff \vdash \llbracket \mathcal{A} \rrbracket [w]$ is provable;
- There is a bijection between the runs and the observations of the proof $\vdash \llbracket \mathcal{A} \rrbracket [w]$. 
\[ \mathcal{L}(\mathcal{A}_1) \subseteq \mathcal{L}(\mathcal{A}_2) \Rightarrow \llbracket A_1 \rrbracket x \vdash \llbracket A_2 \rrbracket x: \]

- We prove the inclusion in $\mu LK^\omega$: 

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

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

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

- 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 $\mathcal{A}_1$;
\[ L(\mathcal{A}_1) \subseteq L(\mathcal{A}_2) \Rightarrow \llbracket A_1 \rrbracket x \vdash \llbracket A_2 \rrbracket x: \]

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

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

- We prove the inclusion in $\mu LK^\infty$:
  - 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 $L(A_1)$ so is in $L(A_2)$. Then one of the run on the right is valid;
\( \mathcal{L}(\mathcal{A}_1) \subseteq \mathcal{L}(\mathcal{A}_2) \Rightarrow \| A_1 \|_x \vdash \| A_2 \|_x \):

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

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

- 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 \( \mu \text{LK}^\omega \);

\( \|A_1\|_x \vdash \|A_2\|_x \Rightarrow \mathcal{L}(A_1) \subseteq \mathcal{L}(A_2) \):
\( \mathcal{L}(A_1) \subseteq \mathcal{L}(A_2) \Rightarrow \|A_1\|_x \vdash \|A_2\|_x: \)

- We prove the inclusion in \( \mu LK^\infty \):
  - 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 \( \mu LK^\omega \);
- then we can build a proof in \( \mu LK \).

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

- 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 \( \| \mathcal{A}_1 \| \) in this branch is valid.

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

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

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

- We prove the inclusion in $\muLK^\infty$:
  - 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 $\mathcal{A}_1$ in this branch is valid.
- then these proofs are regular so are in $\muLK^\omega$;
- then we can build a proof in $\muLK$.

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

- If we prove the inclusion in one of the logics we can prove it in $\muLK^\infty$;
- if $w \in L(\mathcal{A}_1)$ then $\Pi_1 : \vdash \mathcal{A}_1[w]$ and:

\[
\frac{
\Pi_1}{\vdash \mathcal{A}_1[w]} \quad \frac{
\mathcal{A}_1[w] \vdash \mathcal{A}_2[w]
}{\vdash \mathcal{A}_2[w]} \quad \text{cut}
\]
Outline

1. Problems Caused by the Büchi Automata
2. Büchi Automata within the Logics
3. 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.