Equivalence of pointwise and continuous interpretations of first-order logic with linear constraints

Deepak D'Souza
Joint work with Raveeendra Holla and Raj Mohan M.

Department of Computer Science and Automation
Indian Institute of Science, Bangalore.

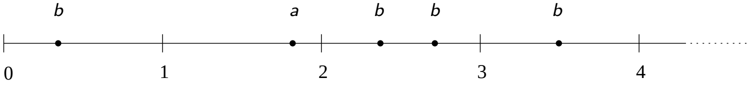
02 February 2010

Outline

- Pointwise and continuous interpretations
- First-Order logic with linear constraints
- From continuous to pointwise.
  - Eliminating a single top-level passive quantifier
  - Eliminating all passive quantifiers.
- Future directions.

Timed words

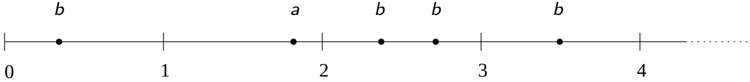
- Timed words [Alur and Dill] are a popular model of real-time behaviours.



- Similar to classical word but each action has a time-stamp.
- Assumption: Time-stamps are progressive.

Quantitative Temporal Logics

- Metric Temporal Logic (MTL) [Koymans 1992, Alur-Feder-Henzinger 1996, Ouaknine-Worrell 2005]
  - aUb "there is a future timepoint at which a b occurs, and till then a occurs."
  - $aU_I b$ "... and the timepoint lies at a distance which lies in the interval I."
  - $\Diamond\varphi \equiv true U\varphi$: "eventually φ."
  - $\Diamond_I\varphi \equiv true U_I\varphi$: "eventually φ at a distance that lies in I."
- Timed Propositional Temporal Logic (TPTL) [Alur-Henzinger 1994].
  - $\Diamond x.(\Diamond y.(a \wedge y = x + 1))$: "There is a future timepoint x and a subsequent timepoint y at which an a occurs and $y = x + 1$."

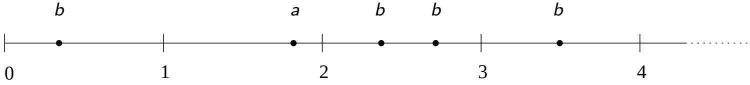


Pointwise vs continuous semantics

Two natural interpretations:

- Pointwise: quantification is over action timepoints in timed word.
- Continuous: quantification is over arbitrary timepoints in timed word.

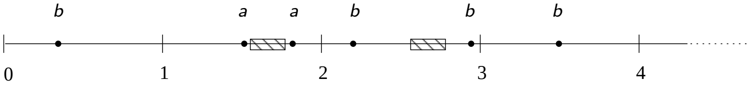
Consider MTL assertion $\Diamond(\Diamond_{[1,1]} a)$ “Eventually there is a timepoint from which we have an action a at distance 1,” on timed word below:



False in pointwise semantics but True in continuous semantics.

Typically pointwise less expressive than continuous

- Pointise MTL is less expressive than Continuous MTL.
    - Property "no insertions" can be expressed in continuous MTL but *not* pointwise MTL.



- Also true for other variants of MTL (MTL_S, MTL_{S_I}, MITL).
- What about TPTL?

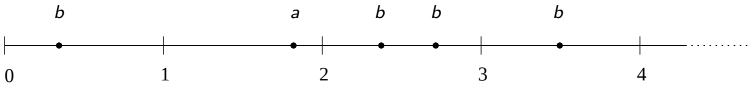
First-Order Logic of linear constraints

- Expressively same as TPTL with "Since" operator.
- Interpreted over timed words.
- $a(x)$: "timepoint x has an a action."
- $x \sim y + c$ where $\sim$ is in $\{<, \leq, =, \geq, >\}$.
- Boolean combinations: $\neg$, $\wedge$, $\vee$.
- First-order quantification: $\exists x \varphi$.

Semantics of $\mathrm{FO}(<,+)$

- Interpreted over timed words.
- $\exists x$ interpreted as
  - "there exists an action point x" (pointwise).
  - "there exists a timepoint x" (continuous).

Example sentence: $\exists x \exists y(a(y) \wedge y = x + 1)$.



Sentence is False in pointwise semantics but True in continuous semantics.

What we show

For a $\mathrm{FO}(<,+)$ sentence φ:

- $L^{pw}(\varphi)$ = set of timed words that satisfy φ in pointwise semantics.
- $L^{c}(\varphi)$ = set of timed words that satisfy φ in continuous semantics.

Theorem

The class of timed languages definable in $\mathrm{FO}(<,+)$ in the pointwise and continuous semantics coincide.

Easy Part: From $\mathrm{FO}^{pw}(<,+)$ to $\mathrm{FO}^{c}(<,+)$

Given φ, find φ' such that $L^{pw}(\varphi) = L^{c}(\varphi')$.
Replace

$$\exists x \psi$$

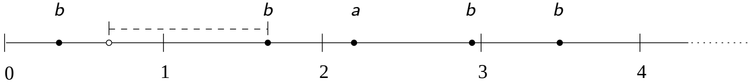
by

$$\exists x \Big(\bigvee_{a \in \Sigma} a(x) \ \wedge \ \psi' \Big).$$

Difficult Part: From $\mathrm{FO}^c(<,+)$ to $\mathrm{FO}^{pw}(<,+)$

Given $\mathrm{FO}^c(<,+)$ sentence:

$$\exists x(\neg a(x) \wedge 0 \leq x \leq 1 \wedge \exists y(b(y) \wedge y = x + 1))$$



A possible equivalent $\mathrm{FO}^{pw}(<,+)$ formula is:

$$\exists y(b(y) \wedge 1 \leq y \leq 2 \wedge \neg \exists x(a(x) \wedge y = x + 1)).$$

Main idea

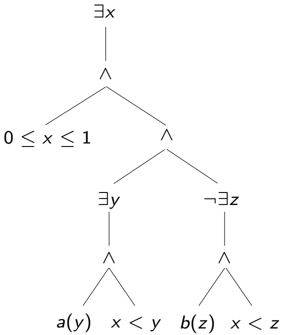
- Go from an $\mathrm{FO}^c(<,+)$ sentence φ to an equivalent actively quantified $\mathrm{FO}^c(<,+)$ sentence φ'.
- Observe that if φ' is actively quantified, then $L^c(\varphi') = L^{pw}(\varphi')$.
- So φ' could be an equivalent $\mathrm{FO}^{pw}(<,+)$ sentence.

Main steps

- First put φ in a normal form.
- Show how to eliminating a single top-level passive quantifier.
- Eliminate all passive quantifiers step by step.

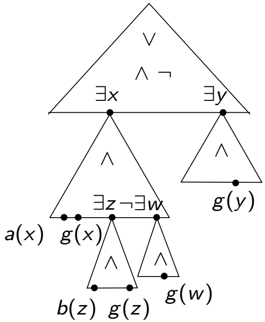
Normal form for $\mathrm{FO}(<,+)$ sentences

- Normal form: Boolean combination of sentences in $\exists$-normal form.
- Example formula in $\exists$-normal form:



Procedure to convert to normal form

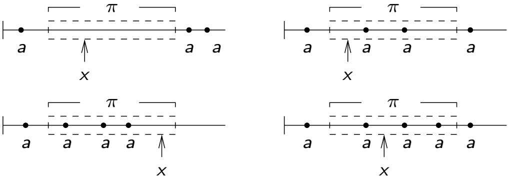
1. Push $\neg$'s downward till $\exists$-nodes or $a(x)$-nodes.
2. Pull $\vee$'s upward (eg. $\exists x(\alpha \vee \beta) \equiv (\exists x\alpha) \vee (\exists y\beta)$).
3. Replace $a(x) \wedge b(x)$ by *false* if $a \neq b$.
4. Replace $\exists x(\neg a(x) \wedge \pi(x) \wedge \alpha)$ by $\psi_1 \vee \psi_2 \vee \psi_3 \vee \psi_4$.



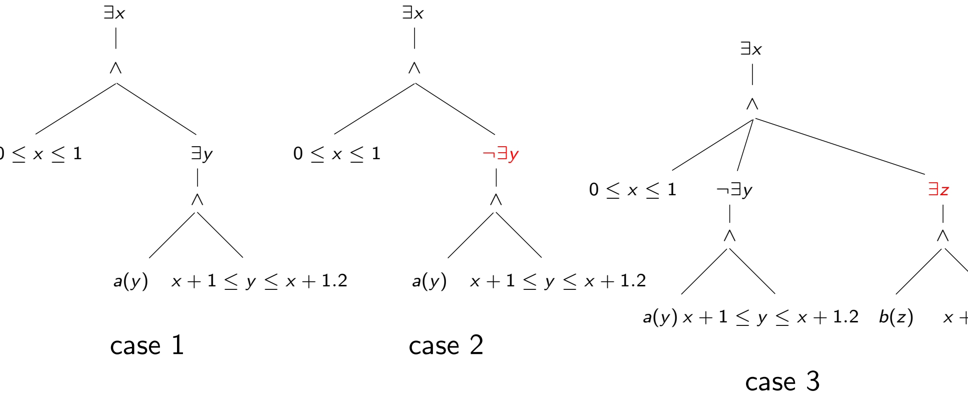
Procedure to convert to normal form

Replace $\exists x(\neg a(x) \land \pi(x) \land \alpha)$ by $\psi_1 \lor \psi_2 \lor \psi_3 \lor \psi_4$, where:

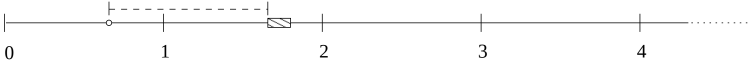
- $\psi_1 = \neg\exists x(a(x) \land \pi(x)) \land \exists x(\pi(x) \ \land \ \alpha)$.
- $\psi_2 = \exists x_l(a(x_l) \land \pi[x_l/x] \land \neg\exists x'(a(x') \land \pi[x'/x] \land x' < x_l) \land \exists x(\pi(x) \land x < x_l \ \land \ \alpha))$.
- Similarly ψ_3, ψ_4.

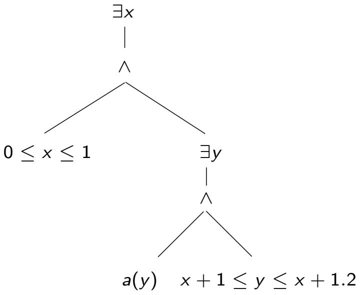


Eliminating a single top-level passive quantifier

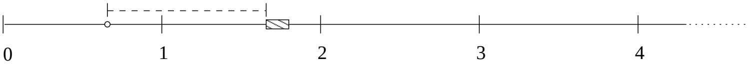


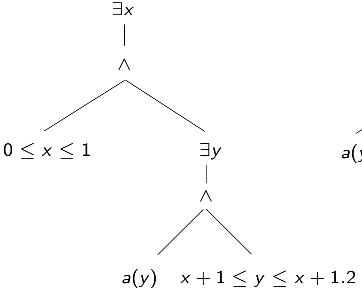
Eliminating a single top-level passive quantifier: case 1

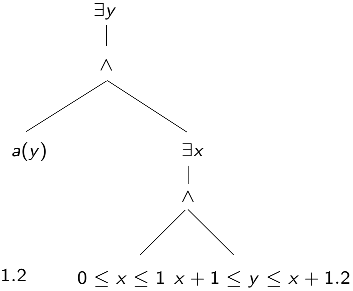




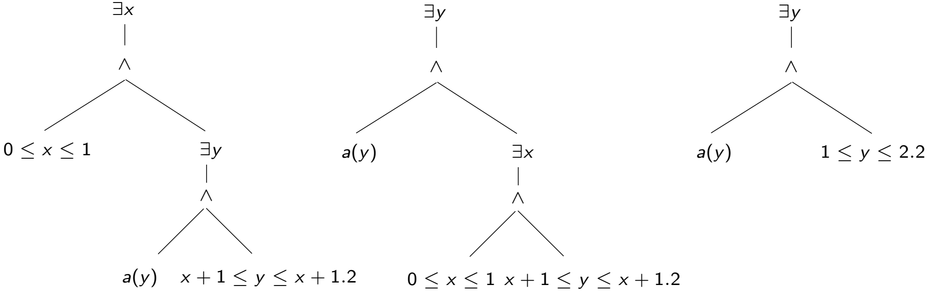
Eliminating a single top-level passive quantifier: case 1



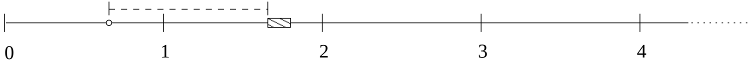


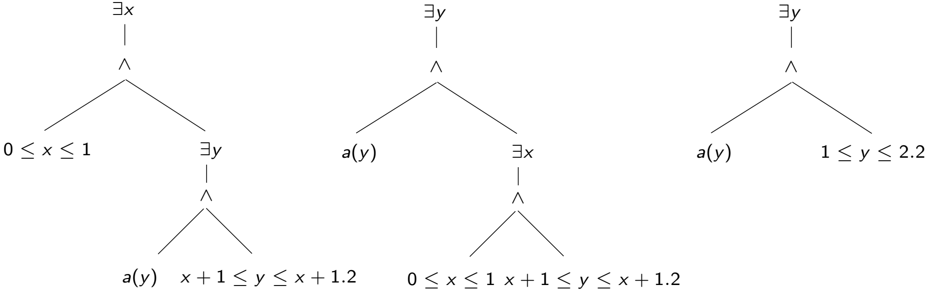


Eliminating a single top-level passive quantifier: case 1



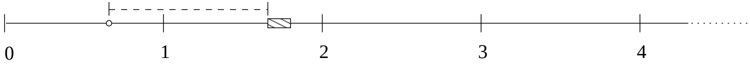
Eliminating a single top-level passive quantifier: case 1

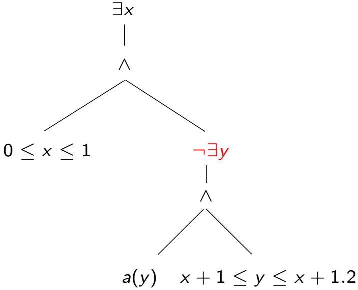




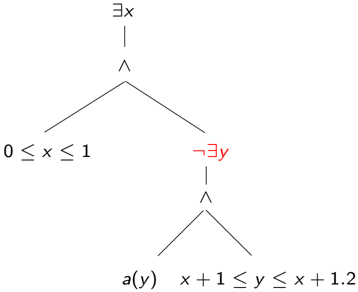
Interval constraint for x: $(0 \leq x \wedge y - 1.2 \leq x) \wedge (x \leq 1 \wedge x \leq y - 1)$.

Eliminating a single top-level passive quantifier: case 2

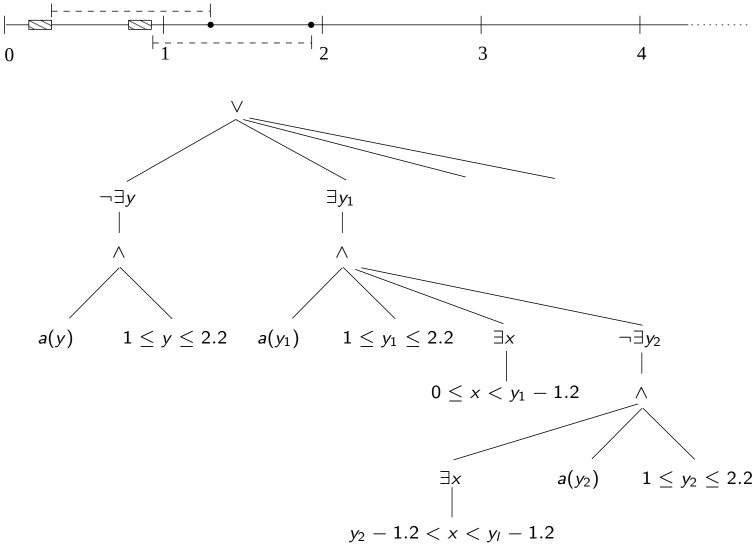




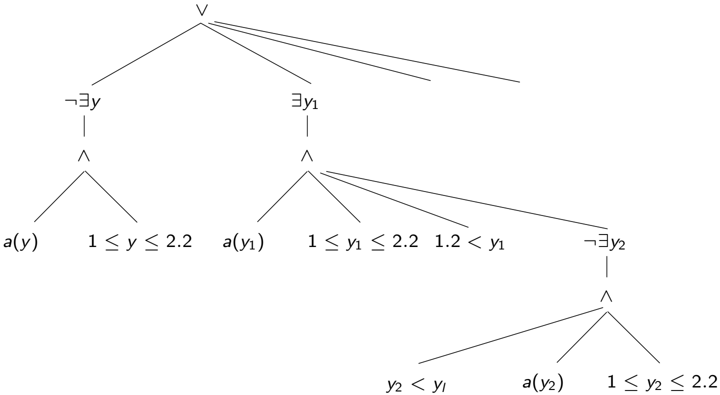
Eliminating a single top-level passive quantifier: case 2



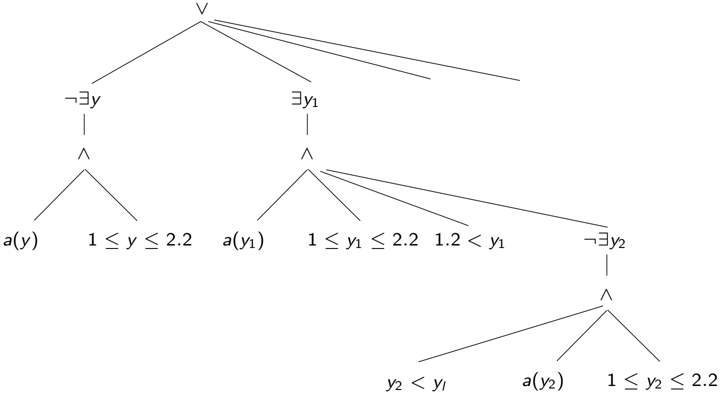
Eliminating a single top-level passive quantifier: case 2



Eliminating a single top-level passive quantifier: case 2

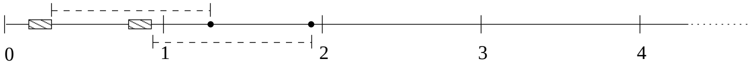


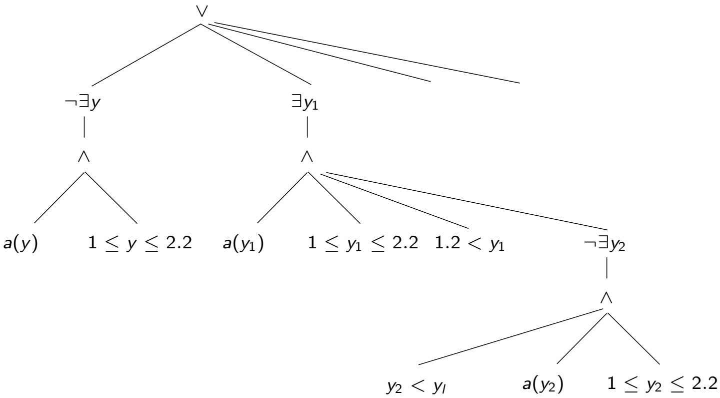
Eliminating a single top-level passive quantifier: case 2



Interval for x for first disjunct: $0 \le x \le 1$.

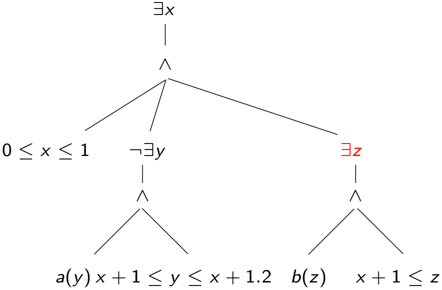
Eliminating a single top-level passive quantifier: case 2



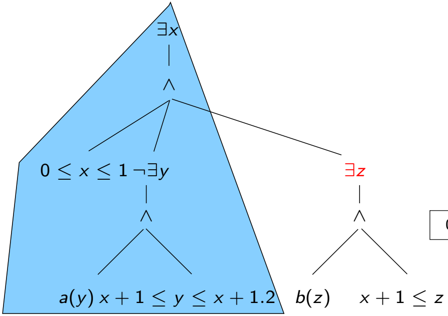


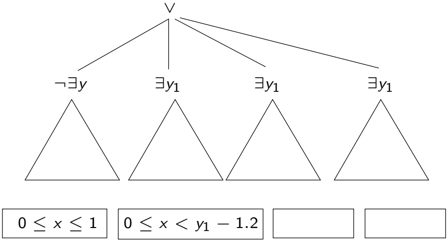
Interval for x for second disjunct: $0 \leq x < y_1 - 1.2$.

Eliminating a single top-level passive quantifier: case 3

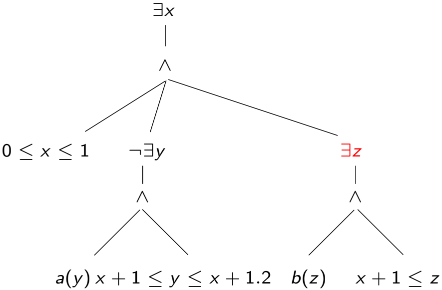


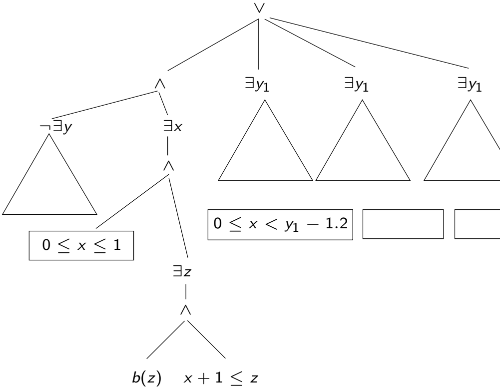
Eliminating a single top-level passive quantifier: case 3



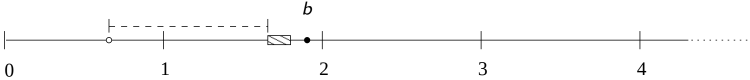


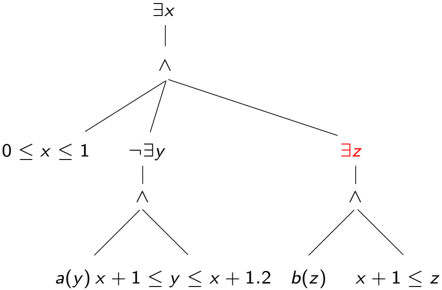
Eliminating a single top-level passive quantifier: case 3

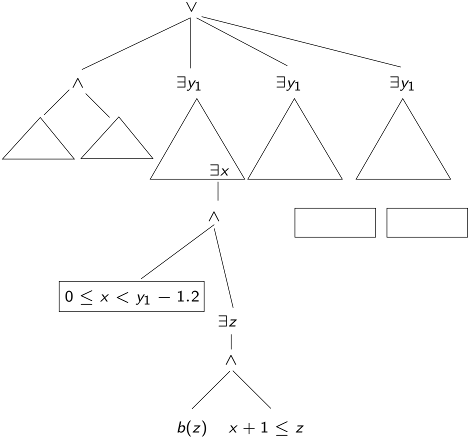




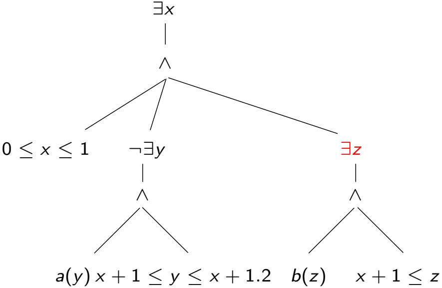
Eliminating a single top-level passive quantifier: case 3





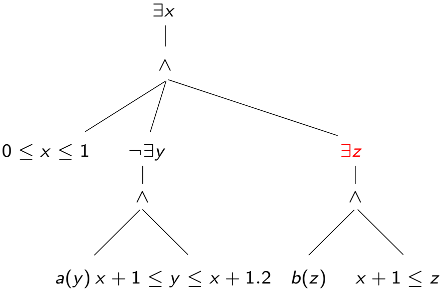


Eliminating a single top-level passive quantifier: case 3



Interval for x for first disjunct: $0 \leq x \leq 1$.

Eliminating a single top-level passive quantifier: case 3

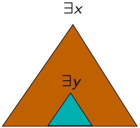


Interval for x for second disjunct: $0 \leq x < y_1 - 1.2$.

Final step: Eliminating all passive quantifiers

Given an $\mathrm{FO}^c(<,+)$ sentence φ:

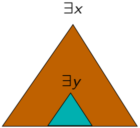
1. Convert to normal form.
2. While there is a passive quantifier node, repeat:
    - Pick a minimal such node.
    - Pull up $\vee' s$ in its subtree (if any)
    - Now each disjunct is in $\exists$-normal form with single top-level passive quantifier. Eliminate this quantifer to get a disjunction of formulas in active normal form.

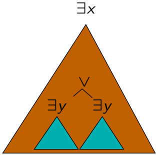


Final step: Eliminating all passive quantifiers

Given an $\mathrm{FO}^c(<,+)$ sentence φ:

1. Convert to normal form.
2. While there is a passive quantifier node, repeat:
   - Pick a minimal such node.
   - Pull up $\vee's$ in its subtree (if any)
   - Now each disjunct is in $\exists$-normal form with single top-level passive quantifier. Eliminate this quantifer to get a disjunction of formulas in active normal form.

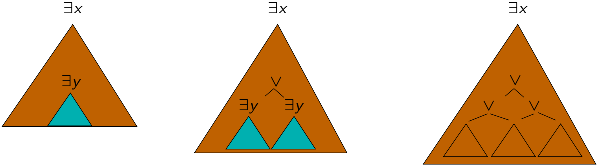




Final step: Eliminating all passive quantifiers

Given an $\mathrm{FO}^c(<,+)$ sentence φ:

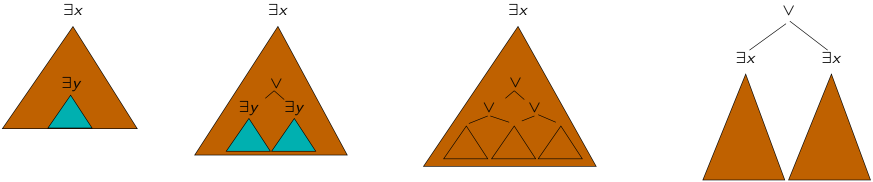
1. Convert to normal form.
2. While there is a passive quantifier node, repeat:
   - Pick a minimal such node.
   - Pull up $\vee' s$ in its subtree (if any)
   - Now each disjunct is in $\exists$-normal form with single top-level passive quantifier. Eliminate this quantifer to get a disjunction of formulas in active normal form.



Final step: Eliminating all passive quantifiers

Given an $\mathrm{FO}^c(<,+)$ sentence φ:

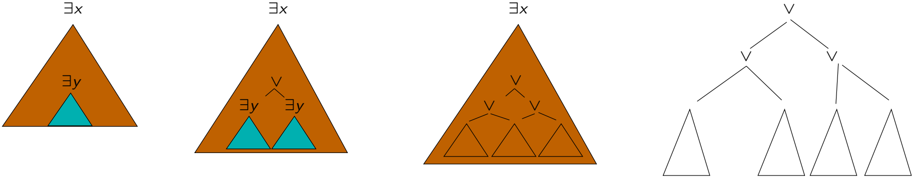
1. Convert to normal form.
2. While there is a passive quantifier node, repeat:
   - Pick a minimal such node.
   - Pull up $\vee's$ in its subtree (if any)
   - Now each disjunct is in $\exists$-normal form with single top-level passive quantifier. Eliminate this quantifer to get a disjunction of formulas in active normal form.



Final step: Eliminating all passive quantifiers

Given an $\mathrm{FO}^c(<,+)$ sentence φ:

1. Convert to normal form.
2. While there is a passive quantifier node, repeat:
   - Pick a minimal such node.
   - Pull up $\vee$'s in its subtree (if any)
   - Now each disjunct is in $\exists$-normal form with single top-level passive quantifier. Eliminate this quantifer to get a disjunction of formulas in active normal form.



Summary

- Shown how to convert an $\mathrm{FO}^c(<,+)$ sentence to an equivalent actively quantified one.
- Gives us equivalence of pointwise and continuous semantics of $\mathrm{FO}(<,+)$.
- Equivalence of pointwise and continuous semantics of TPTL_S follows.
- Some open questions:
  - Compexity?!
  - What about TPTL (without "Since")?