Quantified Linear Arithmetic Satisfiability via Fine-Grained Strategy Improvement

Charlie Murphy
University of Wisconsin-Madison

Zachary Kincaid
Princeton University

July 24, 2024

First Order Logic

Terms

$$t \coloneqq x \mid f\left(t_1, \ldots, t_{rk(f)}\right)$$

$$\varphi, \psi \coloneqq p\left(t_1, \ldots, t_{rk(p)}\right) \mid \forall x{:}\, S.\, \varphi \mid \varphi \wedge \psi$$

$$\mid \neg p\left(t_1, \ldots, t_{rk(p)}\right) \mid \exists x{:}\, S.\, \varphi \mid \varphi \vee \psi$$

Formulas

Negation Normal Form

Linear Rational Arithmetic

Terms

$$t \coloneqq x \mid c \in \mathbb{Q} \mid t_1 + t_2 \mid c \cdot t$$

$$\varphi, \psi \coloneqq \mathrm{t}_1 < \mathrm{t}_2 \mid \forall x{:}\,\mathbb{Q}.\,\varphi \mid \varphi \wedge \psi$$

$$\mid t_1 \leq t_2 \mid \exists x{:}\,\mathbb{Q}.\,\varphi \mid \varphi \vee \psi$$

Formulas

Negation Normal Form

Satisfiability

$$M \models \varphi$$

Model of free variables $X \subseteq dom(M)$

Formula with free variables X

Satisfiability Game

SAT

Controls Angelic Choice
Existential Quantifiers, Disjunction

UNSAT

Controls Demonic Choice
Universal Quantifiers, Conjunction

Satisfiability Game

SAT

UNSAT

$$M \models \varphi$$

Satisfiability Game

SAT

UNSAT

$$G(M, \varphi)$$

State of the game

Satisfiability Game

$$\vDash \forall x, z.\ \ x < z \Rightarrow \exists y.\ x < y < z$$

SAT

UNSAT

Density Property of Rationals

Satisfiability Game

$$G(\emptyset, \forall x, z.\ \ x < z \Rightarrow \exists y.\ x < y < z)$$

SAT

UNSAT

Satisfiability Game

$$G(\emptyset, \forall x, z.\ \ z \leq x \vee (\exists y.\ x < y \wedge y < z)$$

SAT

UNSAT

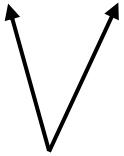
Rewrite to negation normal form

Satisfiability Game

$$G(\emptyset, \forall x, z.\ \ z \leq x \vee (\exists y.\ x < y \wedge y < z)$$

SAT

UNSAT



Controlled by SAT

Satisfiability Game

$$G(\emptyset, \forall x, z.\ \ z \leq x \vee (\exists y.\ x < y \wedge y < z))$$

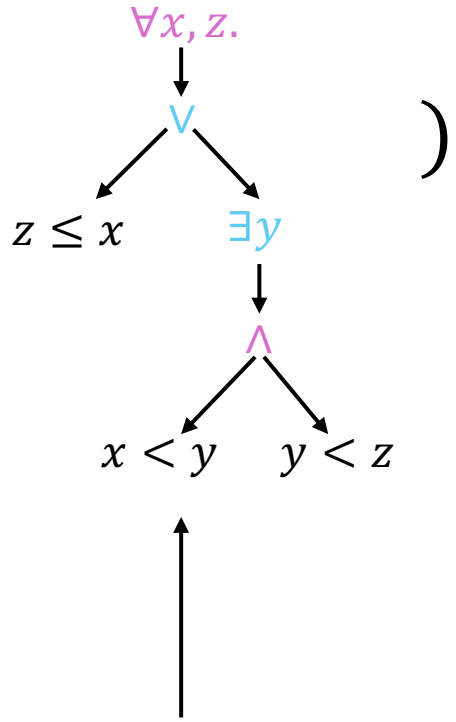
SAT

UNSAT

Controlled by UNSAT

Satisfiability Game

SAT

$G(\emptyset,$  $)$

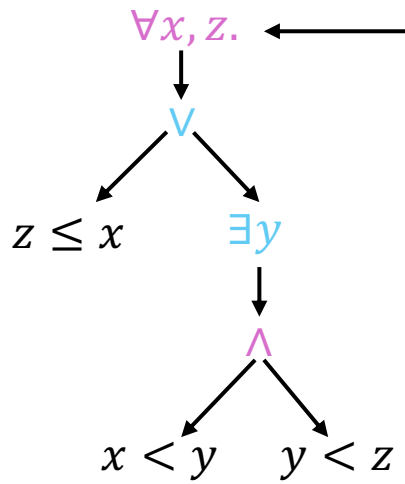
UNSAT

Represent formula as abstract syntax tree (AST)

Satisfiability Game

SAT

$\emptyset \models$



UNSAT

Choose:
$x \mapsto 0$
$z \mapsto 1$

Sequence of actions played so far (by either player).

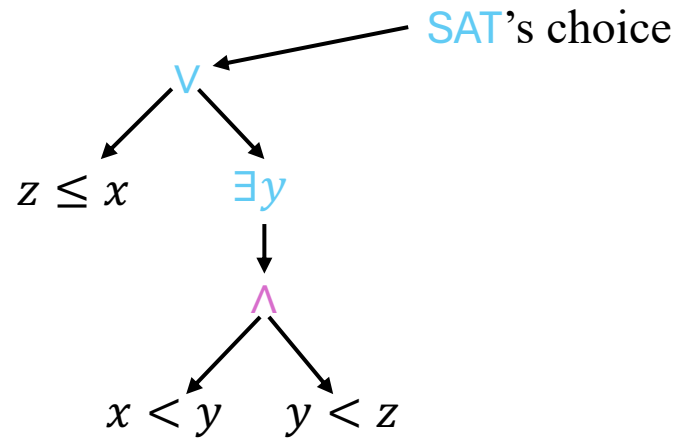
Play: 0 1

Satisfiability Game

SAT

Choose right

$$\{x \mapsto 0, z \mapsto 1\} \models$$



UNSAT

Play: 0 1 R

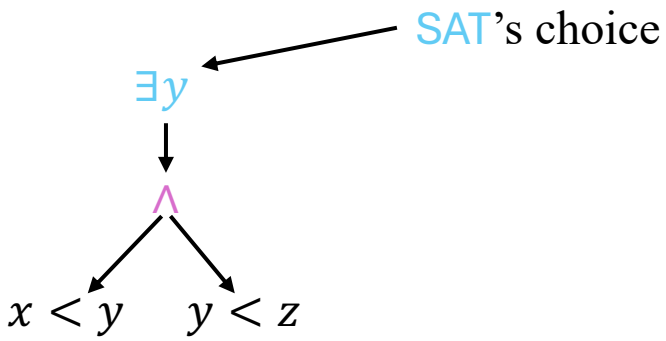
Satisfiability Game

SAT

Choose:

$y \mapsto \frac{1}{2}$

$\{x \mapsto 0, z \mapsto 1\} \models$



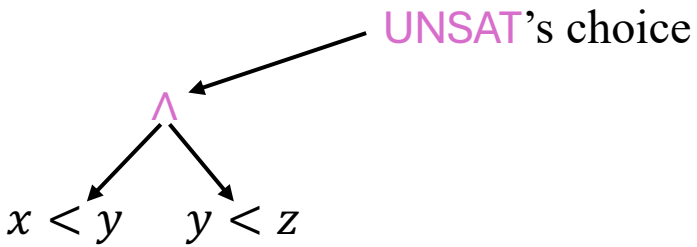
UNSAT

Play: 0 1 R $\frac{1}{2}$

Satisfiability Game

SAT

$$\left\{\begin{array}{l} x \mapsto 0 \\ z \mapsto 1 \\ y \mapsto \frac{1}{2} \end{array}\right\} \models$$



UNSAT

Choose left

Play: 0 1 R $\frac{1}{2}$ L

Satisfiability Game

SAT

$$\left\{\begin{matrix} x \mapsto 0 \\ z \mapsto 1 \\ y \mapsto \frac{1}{2} \end{matrix}\right\} \models \quad x < y$$

UNSAT

The chosen model satisfies the chosen atom.

SAT *wins*.

Play: 0 1 R $\frac{1}{2}$ L

Strategies

f_{SAT} : maps from states of the game that SAT controls to SAT's next move:

- L or R for disjunctions
- a rational value to instantiate an existential quantifier.

Strategies

$$f_{\mathrm{SAT}}\left(\mathrm{M}, \mathrm{z} \leq \mathrm{x} \vee \exists y.\,(x < y \wedge y < z)\right) \stackrel{\text{def}}{=} \text{if } M(z) \leq M(x) \text{ then } L \text{ else } R$$

$$f_{\mathrm{SAT}}\left(\mathrm{M}, \exists y.\,(x < y \wedge y < z)\right) \stackrel{\text{def}}{=} \frac{M(x) + M(z)}{2}$$

f_{SAT} is a *winning* strategy: ⟺ The formula must be satisfiable

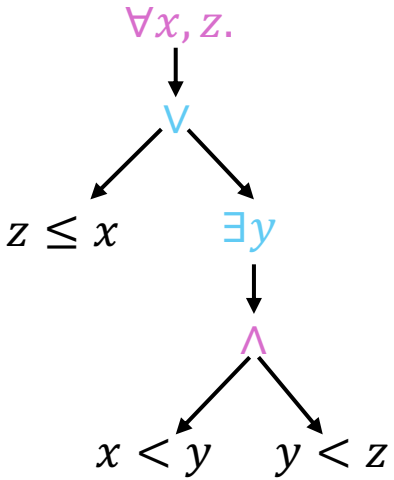
If SAT plays according to f_{SAT}, then SAT will win any play regardless of the choices made by UNSAT.

How do we compute a winning strategy?

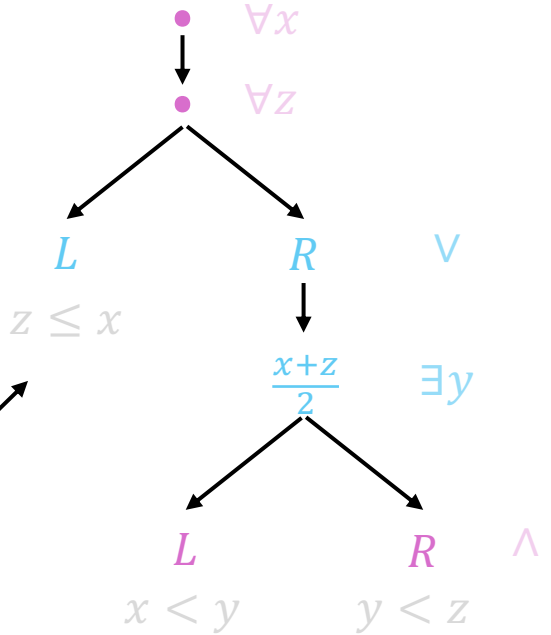
Strategy Skeletons

M: $\emptyset$

φ:



SAT Skeleton S^*

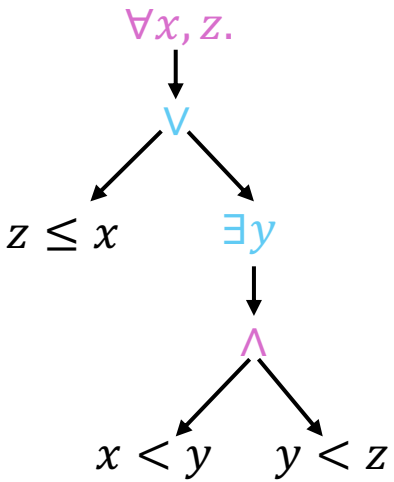


Represents a non-deterministic collection of SAT Strategies whose structure closely follows φ

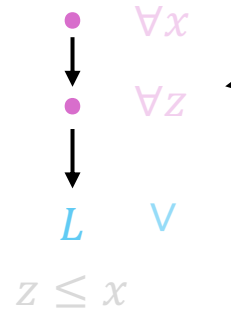
Strategy Improvement

M: $\emptyset$

φ:



SAT Skeleton S_0



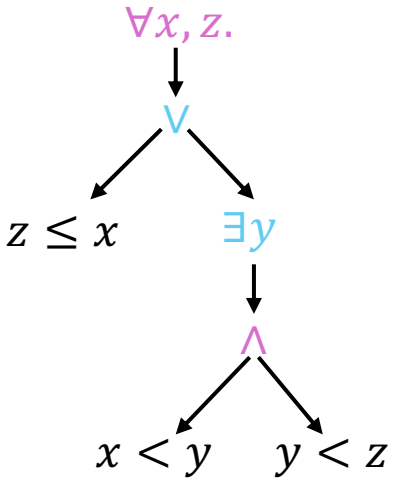
Start with an arbitrary SAT skeleton

Is it winning?

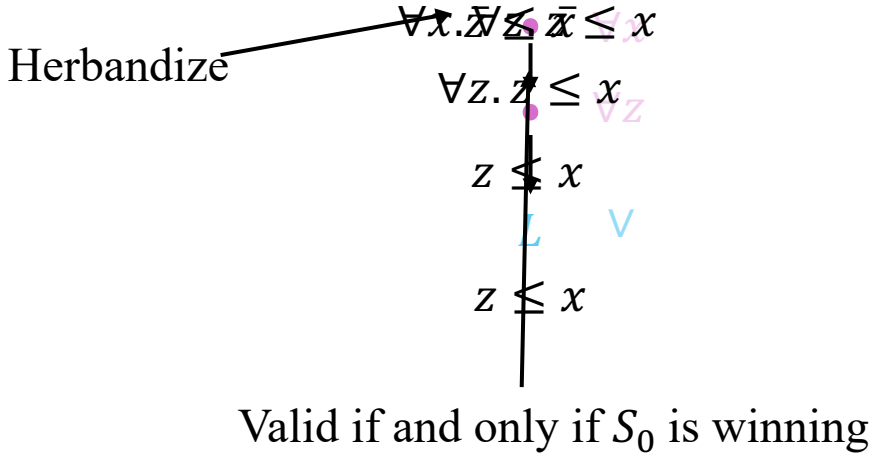
Winning Formula

M: $\emptyset$

φ:



Winning formula of S_0

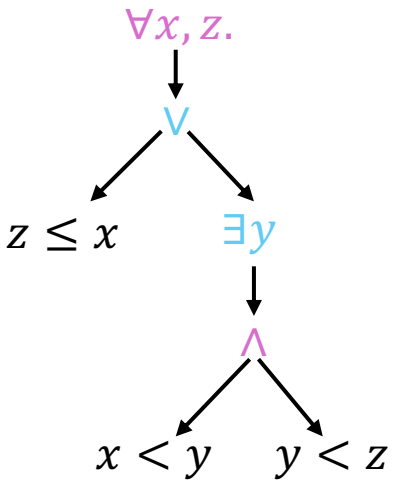


Valid if and only if S_0 is winning

Winning Formula

M: $\emptyset$

φ:



Winning formula of S_0

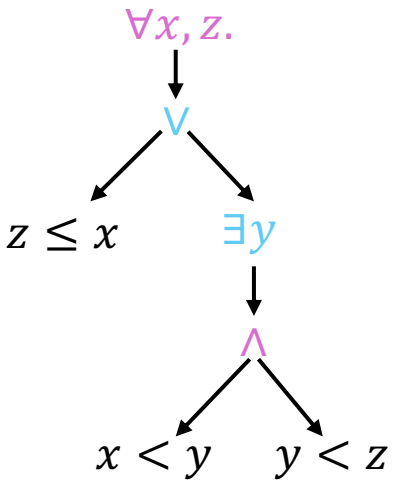
$$M_0 \stackrel{\text{def}}{=} \begin{Bmatrix} \bar{x} \mapsto 0 \\ \bar{z} \mapsto 1 \end{Bmatrix} \models \bar{x} < \bar{z}$$

Satisfiable if and only if S_o is losing

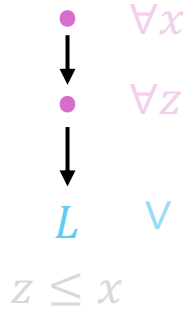
Strategy Improvement

M: $\emptyset$

φ:



SAT Skeleton S_0



S_o has a counter-strategy S_o is not valid

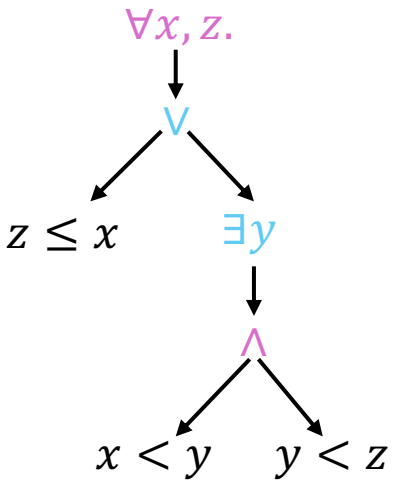
$M_0 \stackrel{\text{def}}{=} \begin{Bmatrix} \bar{x} \mapsto 0 \\ \bar{z} \mapsto 1 \end{Bmatrix} \vDash lose(\varphi, S_0)$

Computing a Counter-Strategy

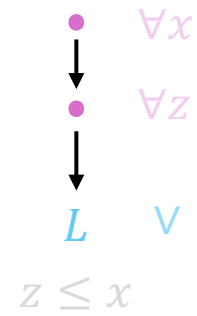
Counter Strategy

M: $\emptyset$

φ:



SAT Skeleton S_0



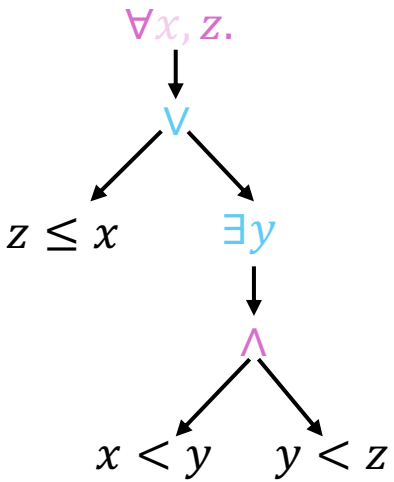
UNSAT Skeleton U_0

$$M_0 \stackrel{\text{def}}{=} \begin{Bmatrix} \bar{x} \mapsto 0 \\ \bar{z} \mapsto 1 \end{Bmatrix} \vDash lose(\varphi, S_0)$$

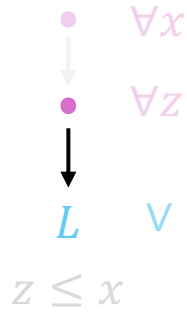
Counter Strategy

M: $\{x \mapsto 0\}$

φ:



SAT Skeleton S_0



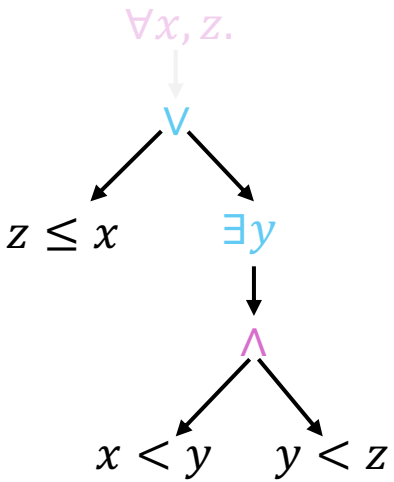
UNSAT Skeleton U_0

$$M_0 \stackrel{\text{def}}{=} \begin{Bmatrix} \bar{x} \mapsto 0 \\ \bar{z} \mapsto 1 \end{Bmatrix} \vDash lose(\varphi, S_0)$$

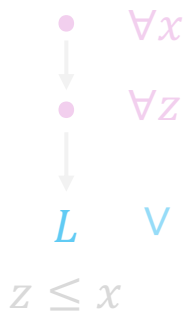
Counter Strategy

M: $\begin{Bmatrix} x \mapsto 0 \\ z \mapsto 1 \end{Bmatrix}$

φ:



SAT Skeleton S_0



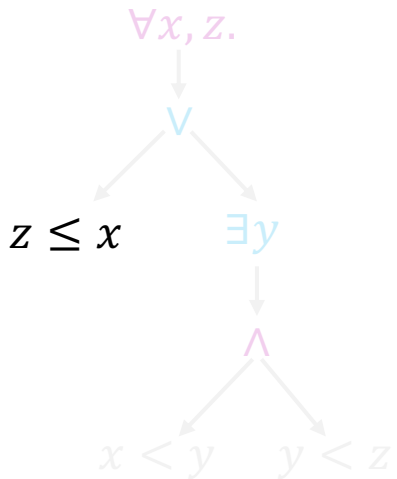
UNSAT Skeleton U_0

$M_0 \stackrel{\text{def}}{=} \begin{Bmatrix} \bar{x} \mapsto 0 \\ \bar{z} \mapsto 1 \end{Bmatrix} \vDash lose(\varphi, S_0)$

Counter Strategy

M: $\begin{Bmatrix} x \mapsto 0 \\ z \mapsto 1 \end{Bmatrix}$

φ:



SAT Skeleton S_0

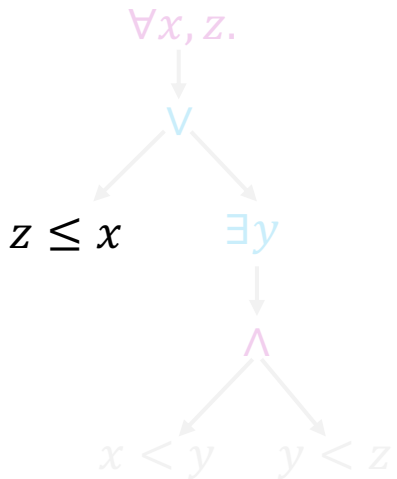
UNSAT Skeleton U_0

$$M_0 \stackrel{\text{def}}{=} \begin{Bmatrix} \bar{x} \mapsto 0 \\ \bar{z} \mapsto 1 \end{Bmatrix} \vDash lose(\varphi, S_0)$$

Counter Strategy

M: $\begin{Bmatrix} x \mapsto 0 \\ z \mapsto 1 \end{Bmatrix}$

φ:



SAT Skeleton S_0

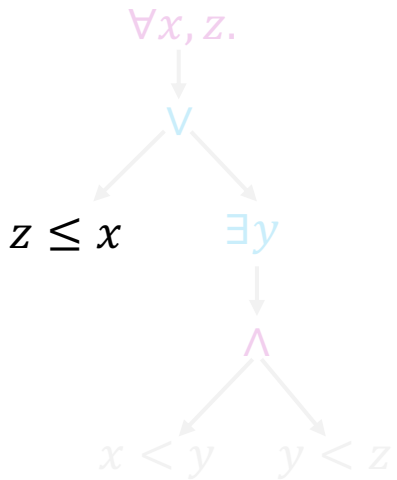
UNSAT Skeleton U_0

$$M_0 \stackrel{\text{def}}{=} \begin{Bmatrix} \bar{x} \mapsto 0 \\ \bar{z} \mapsto 1 \end{Bmatrix} \vDash lose(\varphi, S_0)$$

Counter Strategy

M: $\begin{Bmatrix} x \mapsto 0 \\ z \mapsto 1 \end{Bmatrix}$

φ:



SAT Skeleton S_0

UNSAT Skeleton U_0

$M_0 \stackrel{\text{def}}{=} \begin{Bmatrix} \bar{x} \mapsto 0 \\ \bar{z} \mapsto 1 \end{Bmatrix} \vDash lose(\varphi, S_0)$

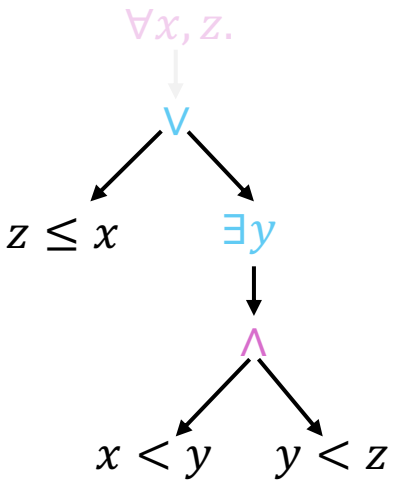
Keeps track of what conditions under which U_o *beats* S_o

$G \stackrel{\text{def}}{=} x < z$

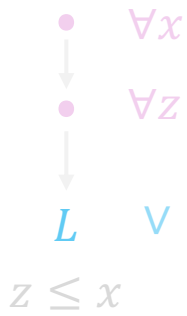
Counter Strategy

M: $\begin{Bmatrix} x \mapsto 0 \\ z \mapsto 1 \end{Bmatrix}$

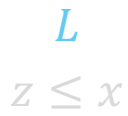
φ:



SAT Skeleton S_0



UNSAT Skeleton U_0



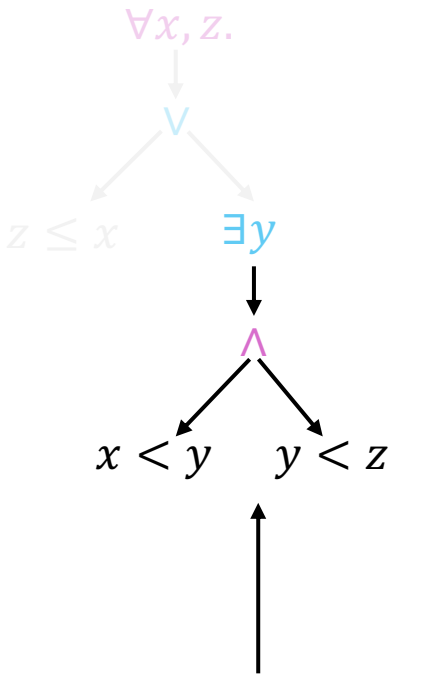
$M_0 \stackrel{\text{def}}{=} \begin{Bmatrix} \bar{x} \mapsto 0 \\ \bar{z} \mapsto 1 \end{Bmatrix} \models lose(\varphi, S_0)$

$G \stackrel{\text{def}}{=} x < z$

Counter Strategy

M: $\begin{Bmatrix} x \mapsto 0 \\ z \mapsto 1 \end{Bmatrix}$

φ:



S_0 has no strategy for this branch

SAT Skeleton S_0

$M_0 \stackrel{\text{def}}{=} \begin{Bmatrix} \bar{x} \mapsto 0 \\ \bar{z} \mapsto 1 \end{Bmatrix} \vDash lose(\varphi, S_0)$

UNSAT Skeleton U_0

L

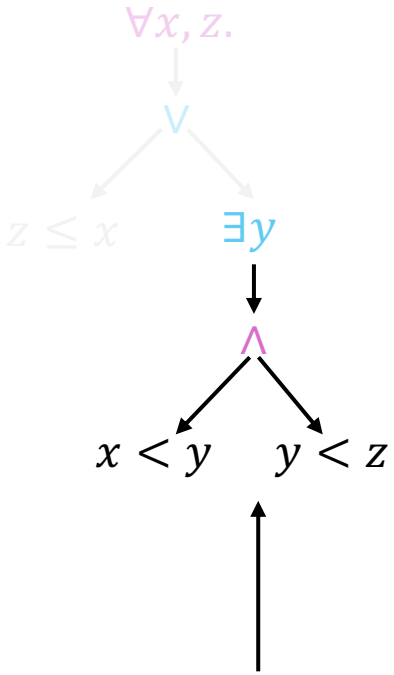
$z \leq x$

$G \stackrel{\text{def}}{=} x < z \wedge$

Counter Strategy

M: $\begin{Bmatrix} x \mapsto 0 \\ z \mapsto 1 \end{Bmatrix}$

φ:

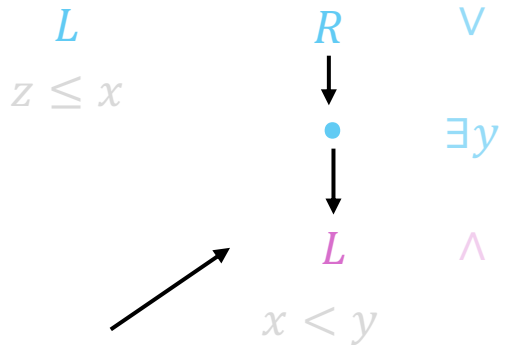


S_0 has no strategy for this branch

SAT Skeleton S_0

$M_0 \stackrel{\text{def}}{=} \begin{Bmatrix} \bar{x} \mapsto 0 \\ \bar{z} \mapsto 1 \end{Bmatrix} \models lose(\varphi, S_0)$

UNSAT Skeleton U_0



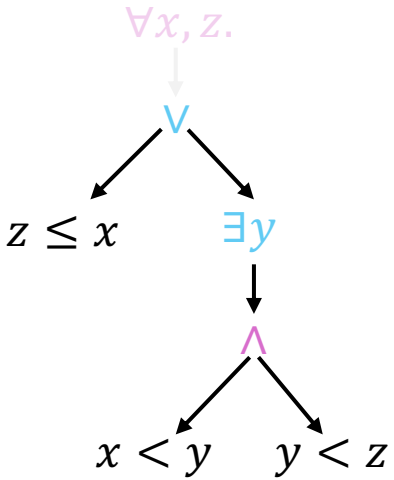
Choose any strategy

$G \stackrel{\text{def}}{=} x < z \wedge true$

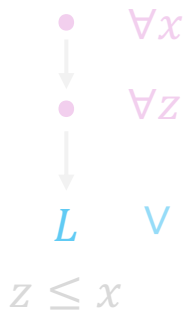
Counter Strategy

M: $\begin{Bmatrix} x \mapsto 0 \\ z \mapsto 1 \end{Bmatrix}$

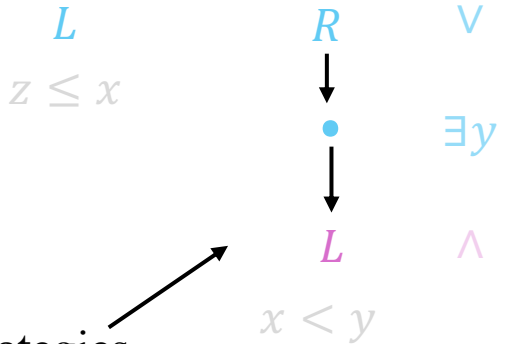
φ:



SAT Skeleton S_0



UNSAT Skeleton U_0



$M_0 \stackrel{\text{def}}{=} \begin{Bmatrix} \bar{x} \mapsto 0 \\ \bar{z} \mapsto 1 \end{Bmatrix} \vDash lose(\varphi, S_0)$

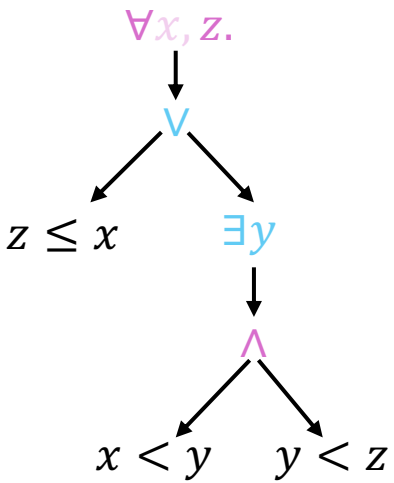
Combine Sub-strategies

$G \stackrel{\text{def}}{=} x < z \wedge true$

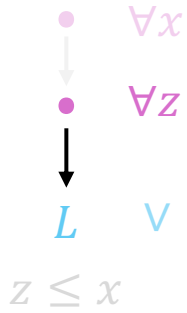
Counter Strategy

M: $\begin{Bmatrix} x \mapsto 0 \\ z \mapsto 1 \end{Bmatrix}$

φ:

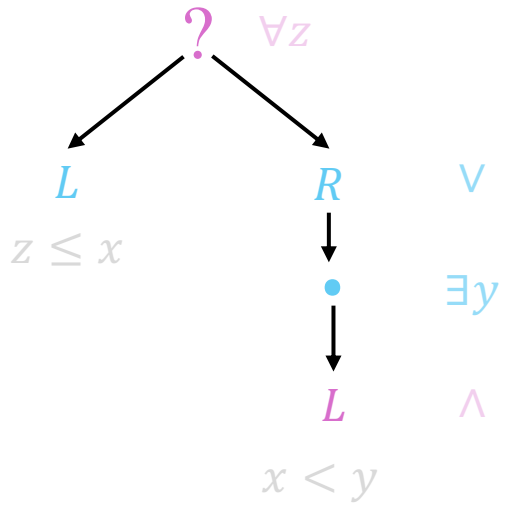


SAT Skeleton S_0



$M_0 \stackrel{\text{def}}{=} \begin{Bmatrix} \bar{x} \mapsto 0 \\ \bar{z} \mapsto 1 \end{Bmatrix} \vDash lose(\varphi, S_0)$

UNSAT Skeleton U_0



$G \stackrel{\text{def}}{=} x < z$

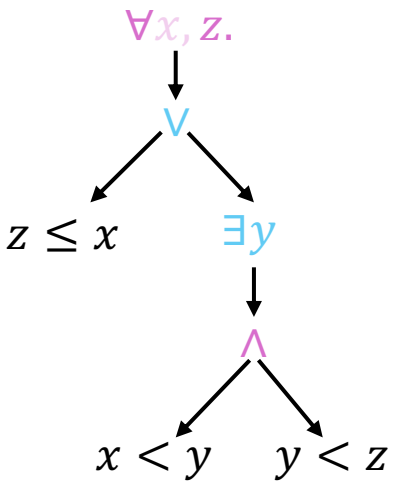
Model-based Term Selection

Term Selection

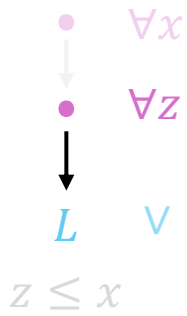
M: $\begin{Bmatrix} x \mapsto 0 \\ z \mapsto 1 \end{Bmatrix}$

At least by model M

φ:



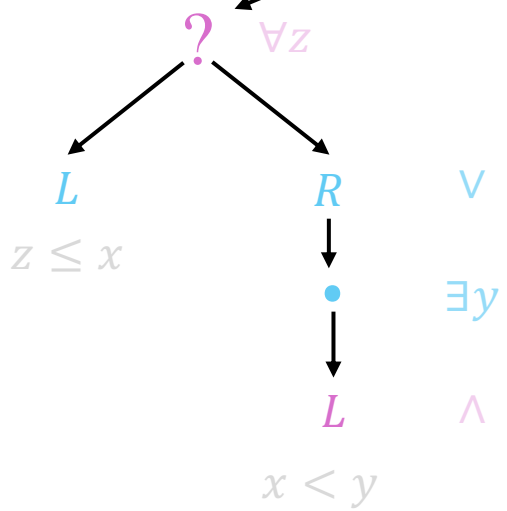
SAT Skeleton S_0



$M_0 \stackrel{\text{def}}{=} \begin{Bmatrix} \bar{x} \mapsto 0 \\ \bar{z} \mapsto 1 \end{Bmatrix} \vDash lose(\varphi, S_0)$

UNSAT Skeleton U_0

Need to find a term to substitute for z



That ensures G Is satisfied

$G \stackrel{\text{def}}{=} x < z$

Term Selection

$t \stackrel{\text{def}}{=} select(x, M, G)$

Find a t such that:

- $\text{FV}(t) \subseteq FV(G) \setminus \{x\}$
- If $M \vDash G$ then $M \vDash G[x \mapsto t]$

LRA Term Selection

$$select(x, M, G) \stackrel{\text{def}}{=} \begin{cases} eq(x, M, G) \\ \frac{1}{2}(glb(x,M,G)+lub(x,M,G)) \\ glb(x, M, G) + 1 \\ lub(x, M, G) - 1 \\ 0 \end{cases}$$

Pick t such that M $\models x = t$ and $x = t$ is a sub-formula of G

LRA Term Selection

$$select(x, M, G) \stackrel{\text{def}}{=} \begin{cases} eq(x, M, G) \\ \frac{1}{2}(glb(x,M,G)+lub(x,M,G)) \\ glb(x, M, G) + 1 \\ lub(x, M, G) - 1 \\ 0 \end{cases}$$

Pick t such that

- M $\vDash t < x$
- $t < x$ is a sub-formula of G
- There is no t' such that
  - M $\vDash t' < x$
  - $t' < x$ is a sub-formula of G
  - $M \vDash t < t'$

LRA Term Selection

$$select(x, M, G) \stackrel{\text{def}}{=} \begin{cases} eq(x, M, G) \\ \frac{1}{2}(glb(x,M,G)+lub(x,M,G)) \\ glb(x, M, G) + 1 \\ lub(x, M, G) - 1 \\ 0 \end{cases}$$

Pick t such that

- M $\models x < t$
- $x < t$ is a sub-formula of G
- There is no t' such that
  - M $\models x < t'$
  - $x < t'$ is a sub-formula of G
  - $M \models t' < t$

LRA Term Selection

$$select(x, M, G) \stackrel{\text{def}}{=} \begin{cases} eq(x, M, G) \\ \frac{1}{2}(glb(x,M,G)+lub(x,M,G)) \\ glb(x, M, G) + 1 \\ lub(x, M, G) - 1 \\ 0 \end{cases}$$

0 ← x does not appear in G

LRA Term Selection

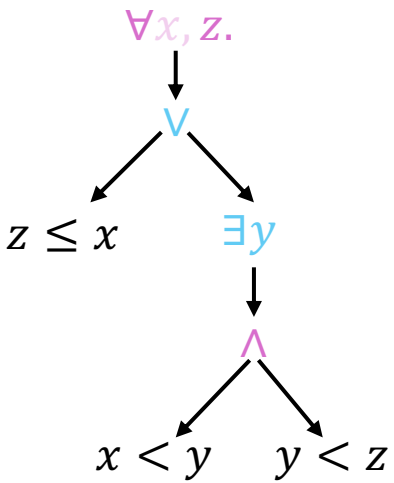
$$select\left(z, \begin{Bmatrix} x \mapsto 0 \\ z \mapsto 1 \end{Bmatrix}, x < z\right) \stackrel{\text{def}}{=} x + 1$$

Computing a Counter Strategy

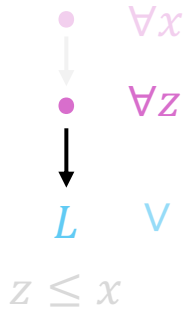
Counter Strategy

M: $\begin{Bmatrix} x \mapsto 0 \\ z \mapsto 1 \end{Bmatrix}$

φ:

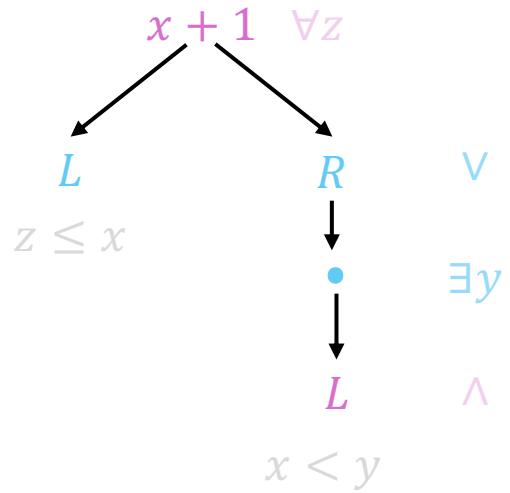


SAT Skeleton S_0



$M_0 \stackrel{\text{def}}{=} \begin{Bmatrix} \bar{x} \mapsto 0 \\ \bar{z} \mapsto 1 \end{Bmatrix} \vDash lose(\varphi, S_0)$

UNSAT Skeleton U_0

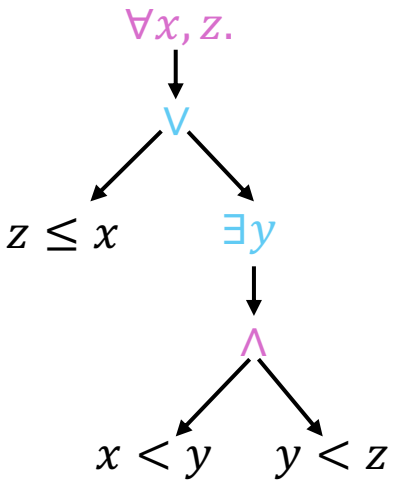


$G \stackrel{\text{def}}{=} x < z$

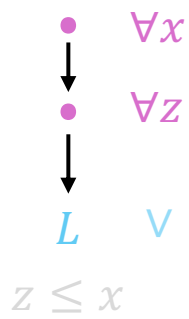
Counter Strategy

M: $\{x \mapsto 0\}$

φ:

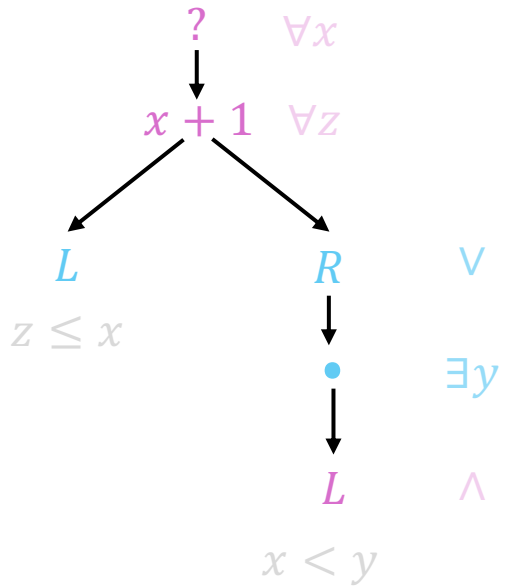


SAT Skeleton S_0



$$M_0 \stackrel{\text{def}}{=} \begin{Bmatrix} \bar{x} \mapsto 0 \\ \bar{z} \mapsto 1 \end{Bmatrix} \vDash lose(\varphi, S_0)$$

UNSAT Skeleton U_0

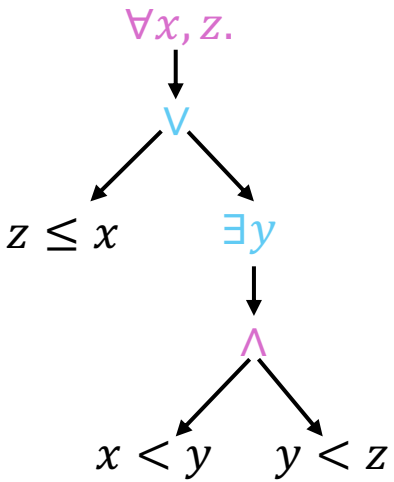


$$G \stackrel{\text{def}}{=} x < x + 1$$

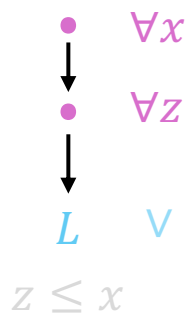
Counter Strategy

M: $\emptyset$

φ:

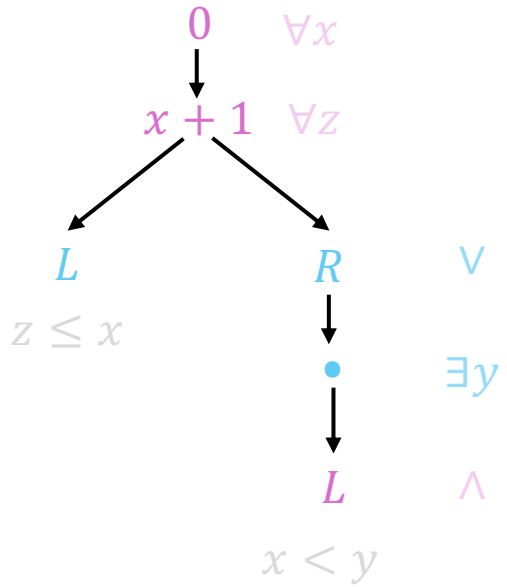


SAT Skeleton S_0



$$M_0 \stackrel{\text{def}}{=} \begin{Bmatrix} \bar{x} \mapsto 0 \\ \bar{z} \mapsto 1 \end{Bmatrix} \vDash lose(\varphi, S_0)$$

UNSAT Skeleton U_0



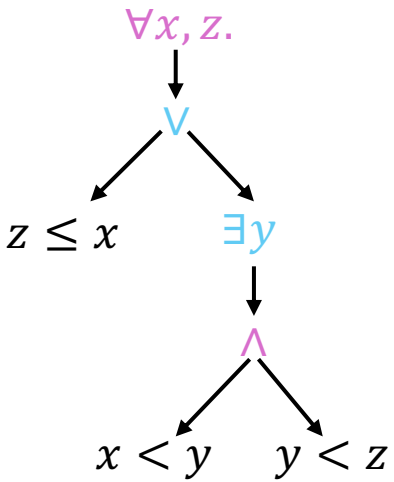
$$G \stackrel{\text{def}}{=} true$$

Computing a winning skeleton

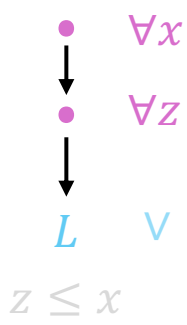
Strategy Improvement

M: $\emptyset$

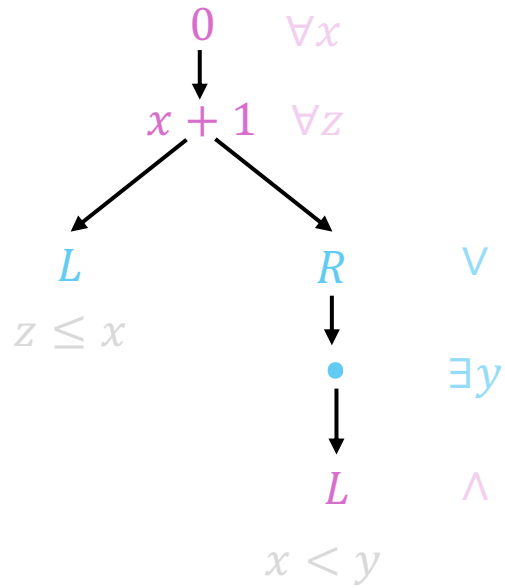
φ:



SAT Skeleton S_0



UNSAT Skeleton U_0

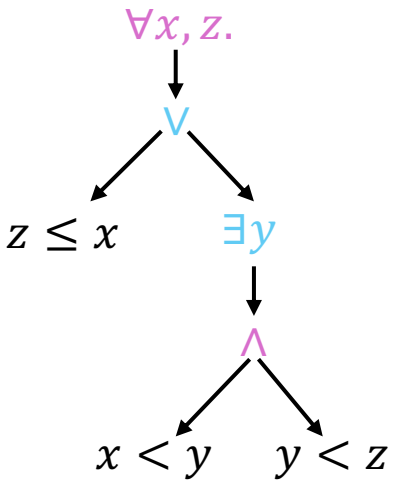


U_0 *beats* S_0

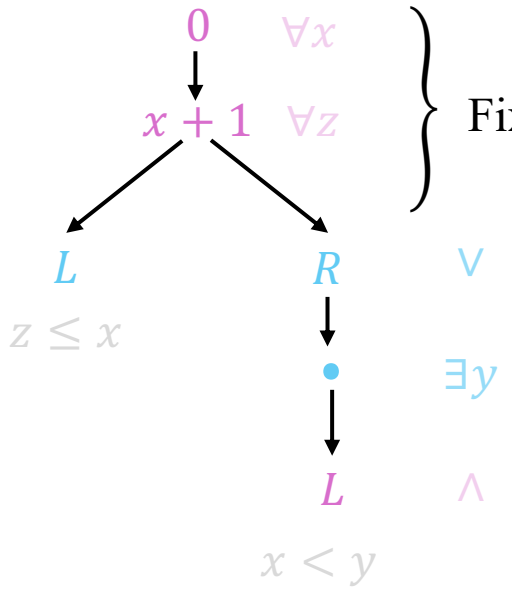
Strategy Improvement

M: $\emptyset$

φ:



UNSAT Skeleton U_0

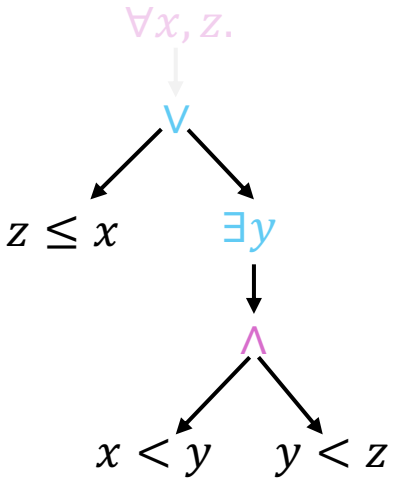


Fix the leading moves of U_0

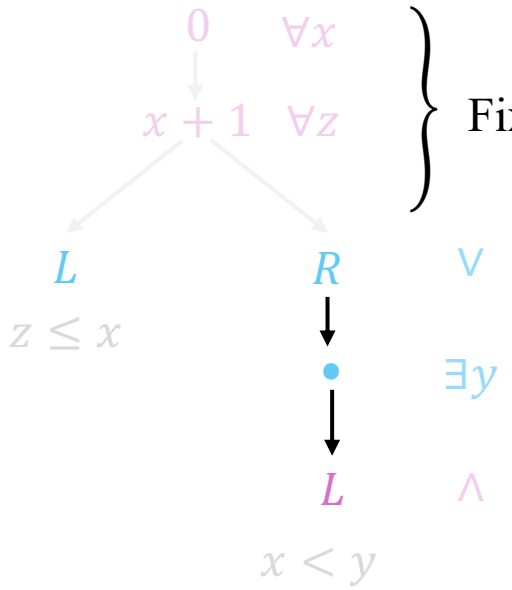
Strategy Improvement

M: $\begin{Bmatrix} x \mapsto 0 \\ z \mapsto 1 \end{Bmatrix}$

φ:



UNSAT Skeleton U_0

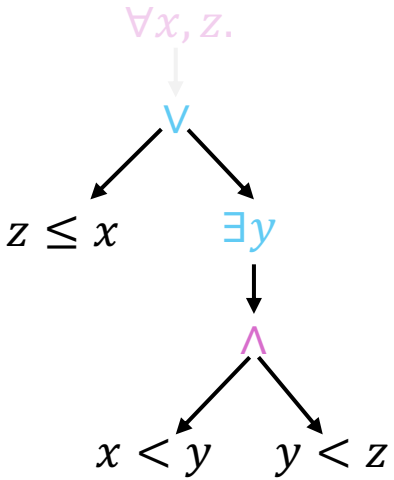


Fix the leading moves of U_0

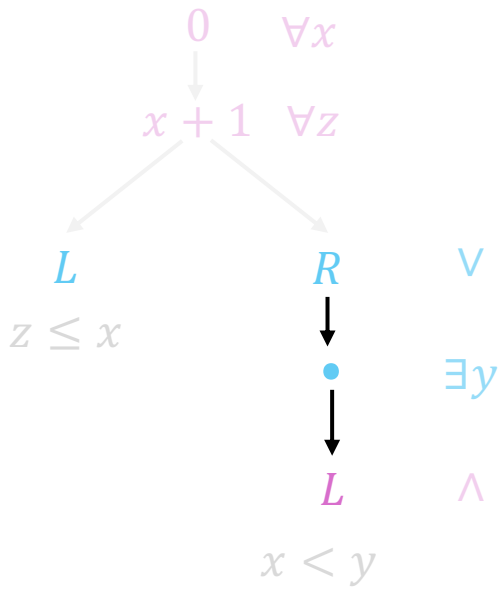
Strategy Improvement

M: $\{x \mapsto 0, z \mapsto 1\}$

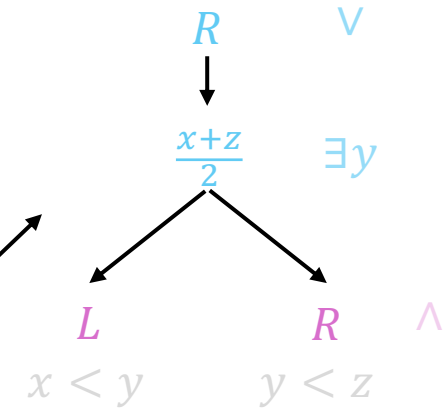
φ:



UNSAT Skeleton U_0



SAT Skeleton S_1'

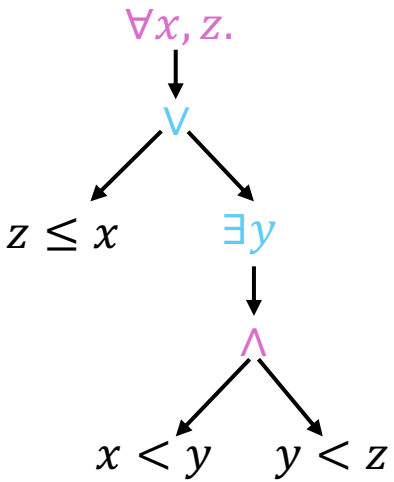


Swap roles and recursively solve the resulting sub-game

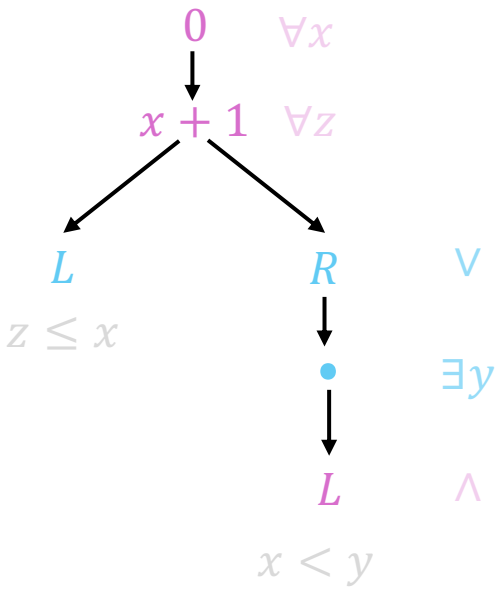
Strategy Improvement

M: $\emptyset$

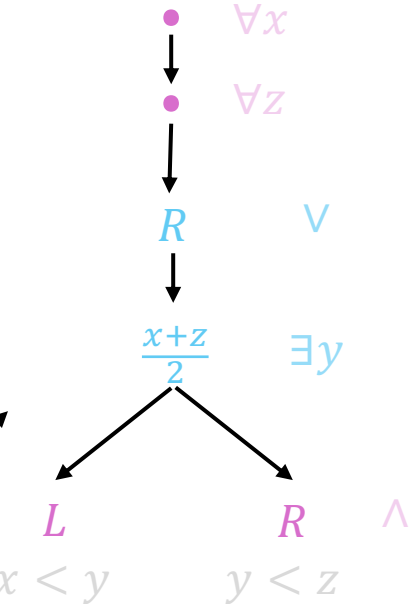
φ:



UNSAT Skeleton U_0



SAT Skeleton S_1'



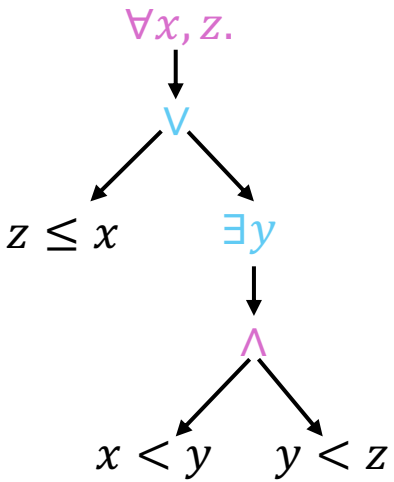
Generalizes U_0 strategy

S_1' beats U_0 strategy

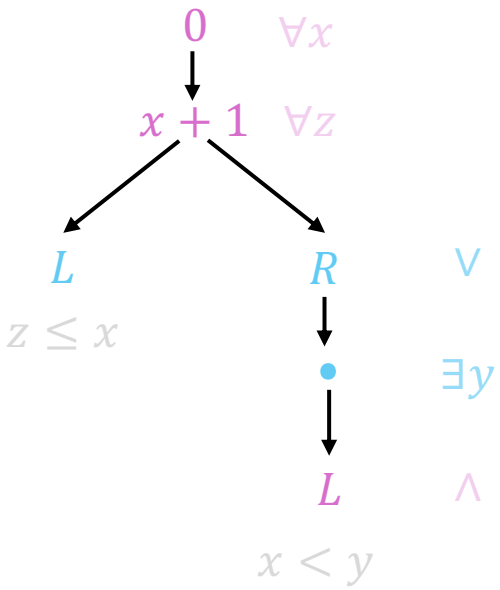
Strategy Improvement

M: $\emptyset$

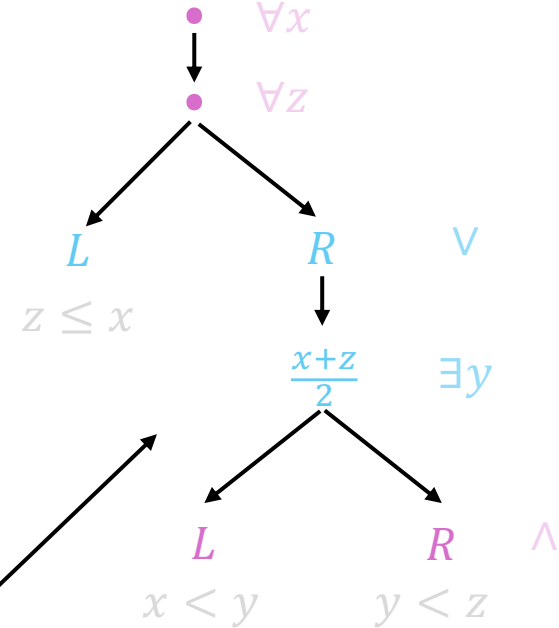
φ:



UNSAT Skeleton U_0



SAT Skeleton S_1

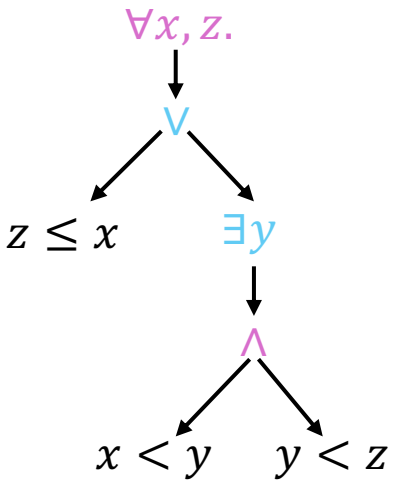


Combine with S_0

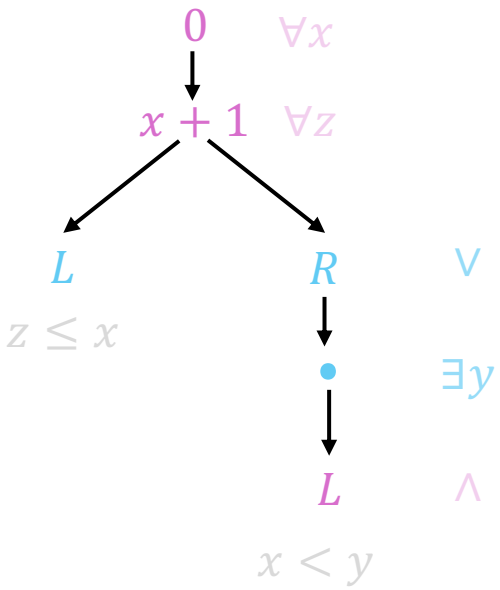
Strategy Improvement

M: $\emptyset$

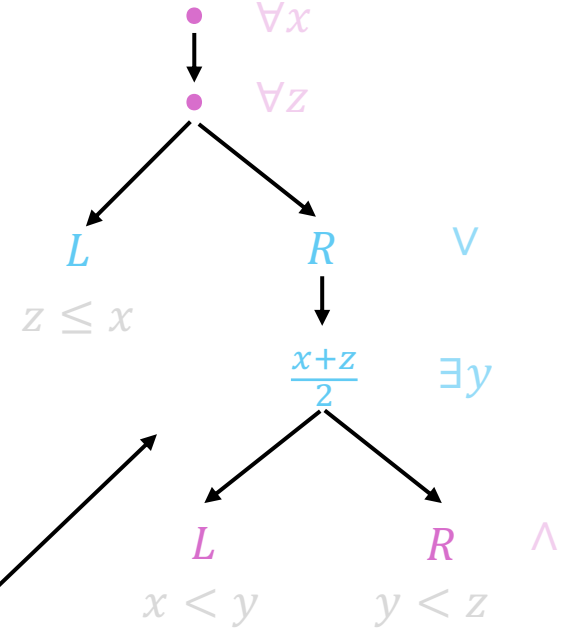
φ:



UNSAT Skeleton U_0



SAT Skeleton S_1

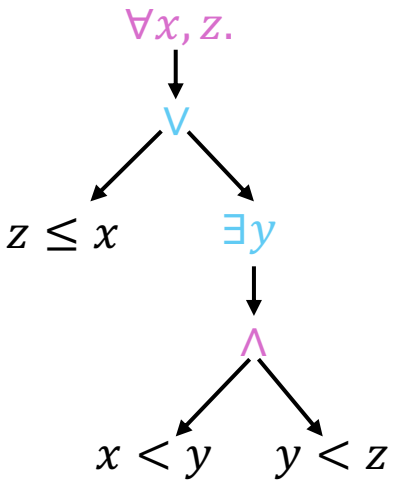


$S_0 \subseteq S_1$ and S_1 beats U_0

Strategy Improvement

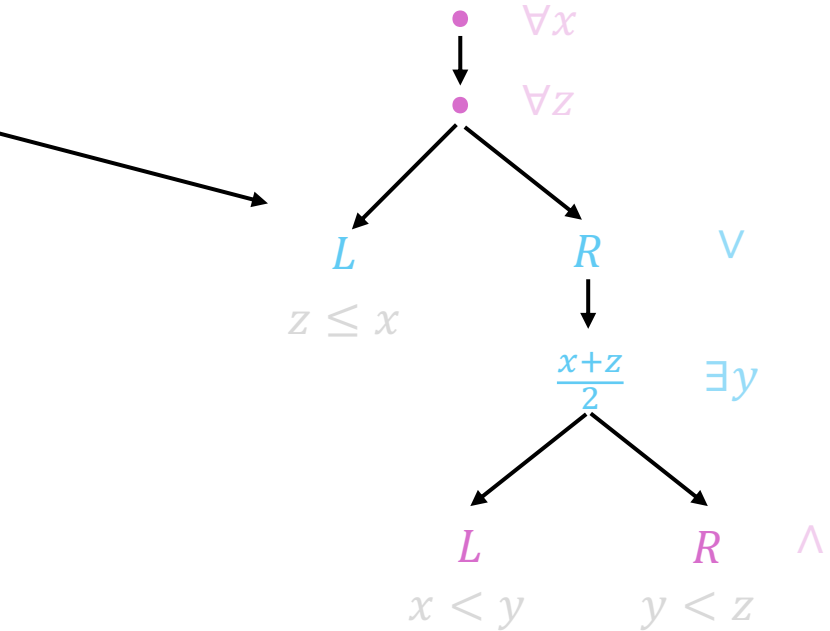
M: $\emptyset$

φ:



S_1 is winning

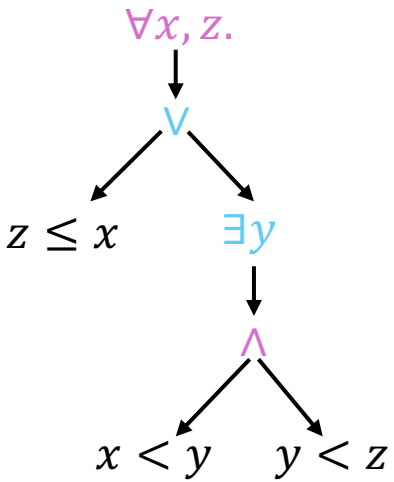
SAT Skeleton S_1



Strategy Improvement

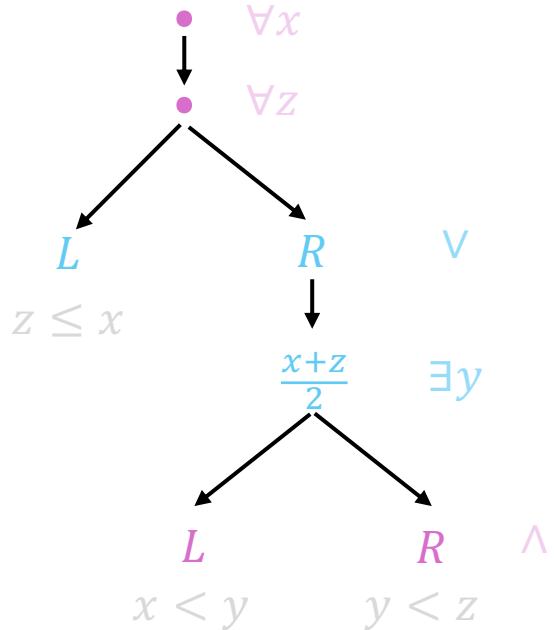
M: $\emptyset$

φ:



Some strategy conforming to S_1 is winning

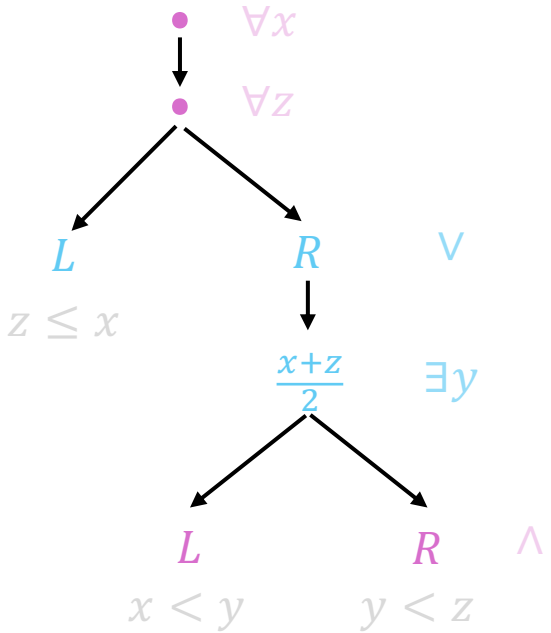
SAT Skeleton S_1

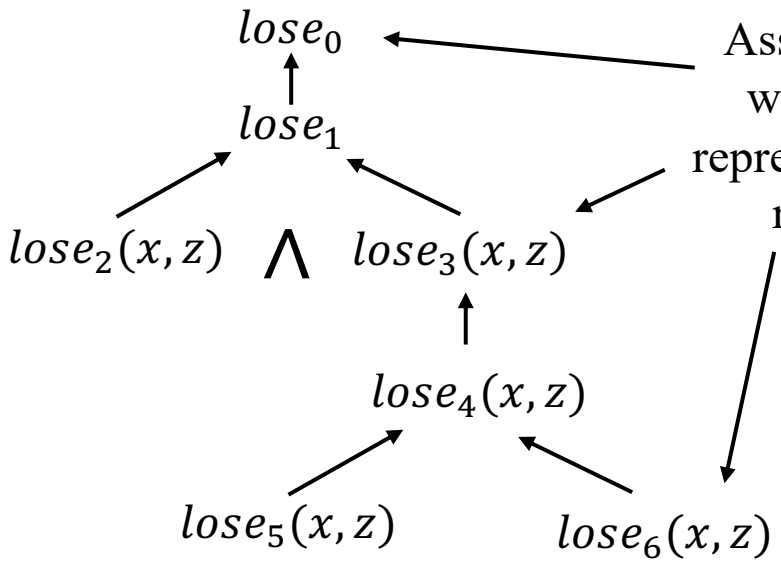


Computing a strategy

Strategy Synthesis

Winning SAT Skeleton

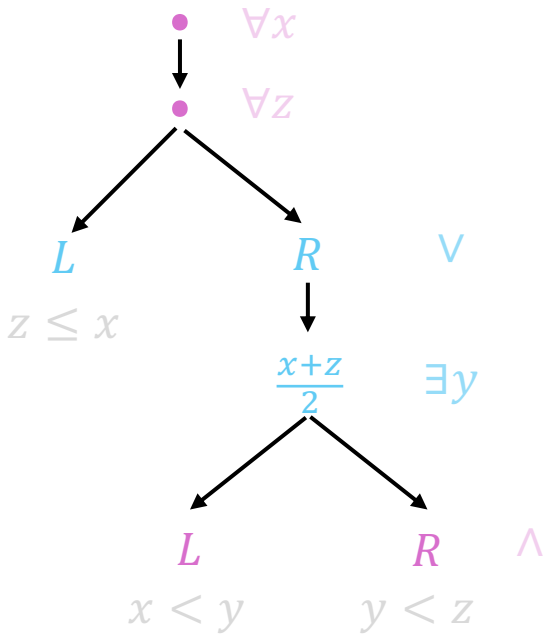


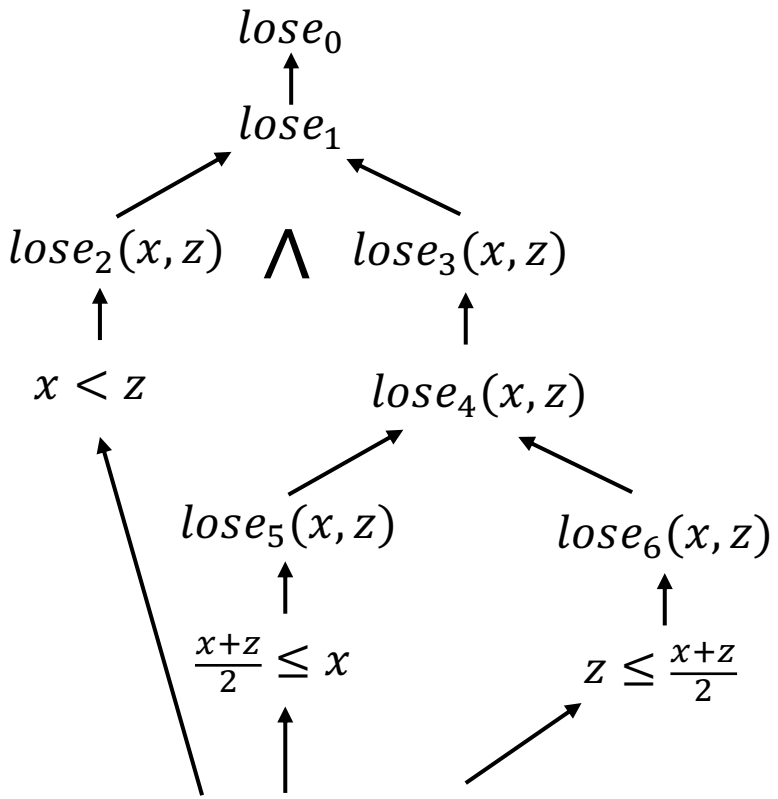


Associate each node of skeleton with an uninterpreted relation representing when the sub-skeleton rooted at the node is losing

Strategy Synthesis

Winning SAT Skeleton

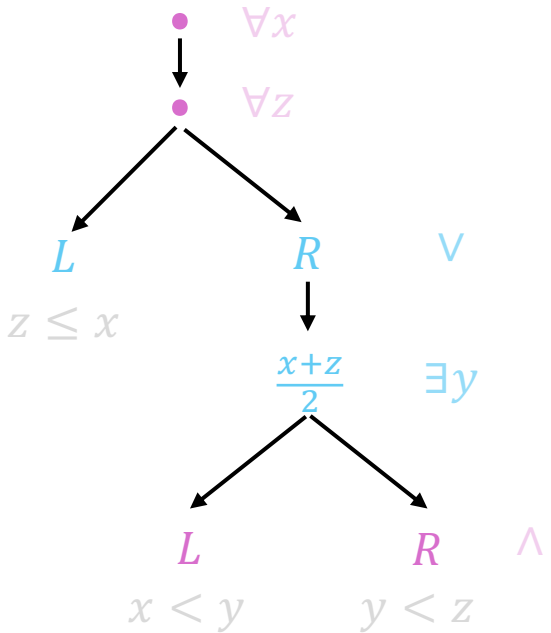


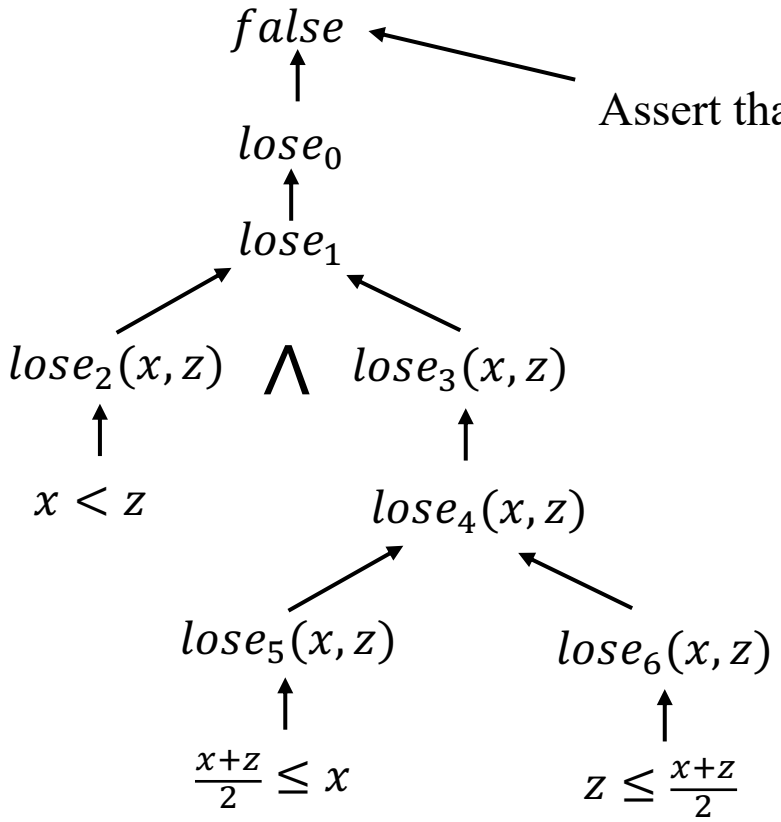


Label each atomic skeleton with the dual of its atom

Strategy Synthesis

Winning SAT Skeleton

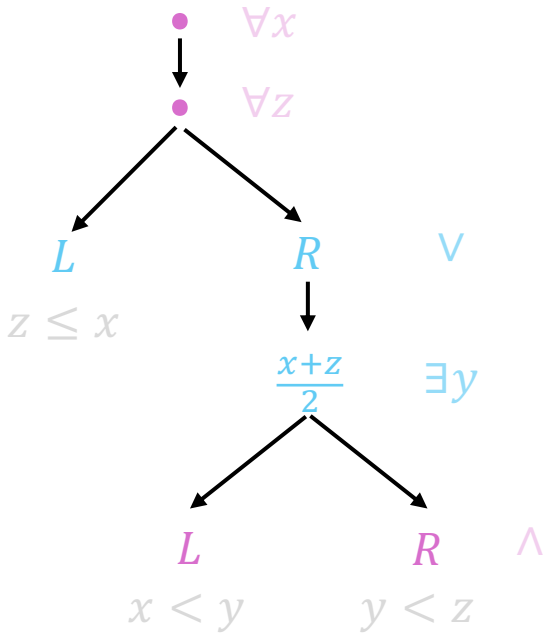


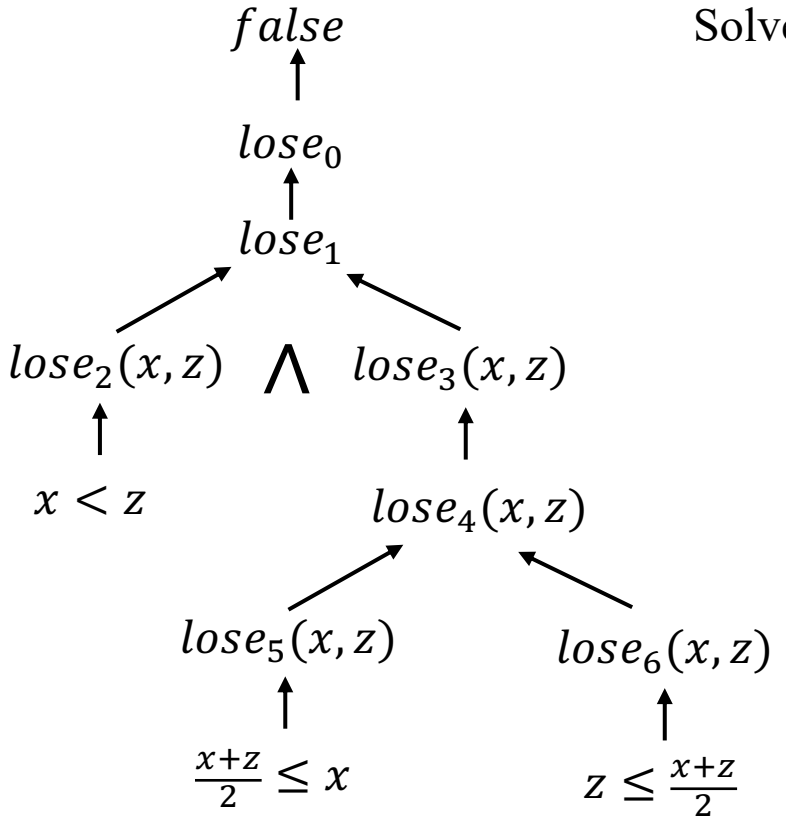


Assert that the root is not losing

Strategy Synthesis

Winning SAT Skeleton





Solve the system of constraints:

$lose_0 \stackrel{\text{def}}{=} false$

$lose_1 \stackrel{\text{def}}{=} false$

$lose_2(x,z) \stackrel{\text{def}}{=} x < z$

$lose_3(x,z) \stackrel{\text{def}}{=} z \leq x$

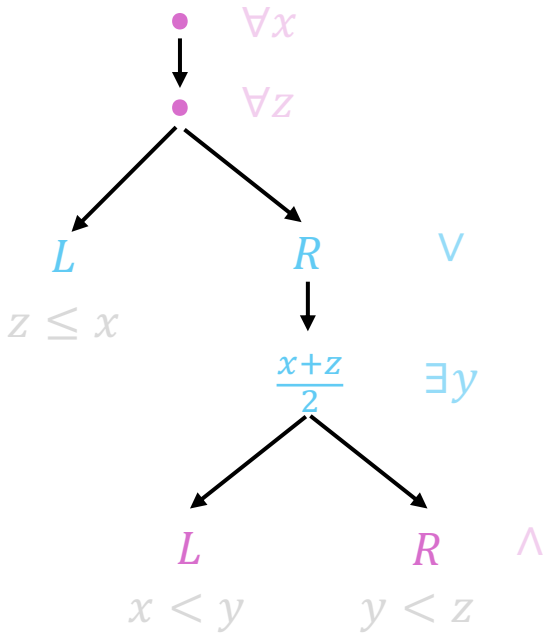
$lose_4(x,z) \stackrel{\text{def}}{=} z \leq x$

$lose_5(x,z) \stackrel{\text{def}}{=} z \leq x$

$lose_6(x,z) \stackrel{\text{def}}{=} z \leq x$

Strategy Synthesis

Winning SAT Skeleton



$\neg lose_2(x,z)$

$f_{..}(x,z) \stackrel{\text{def}}{=} \text{if } z \leq x \; L \text{ else } R$

$f_{..R}(x,z) \stackrel{\text{def}}{=} \frac{x+z}{2}$

Solve the system of constraints:

$lose_0 \stackrel{\text{def}}{=} false$

$lose_1 \stackrel{\text{def}}{=} false$

$lose_2(x,z) \stackrel{\text{def}}{=} x < z$

$lose_3(x,z) \stackrel{\text{def}}{=} z \leq x$

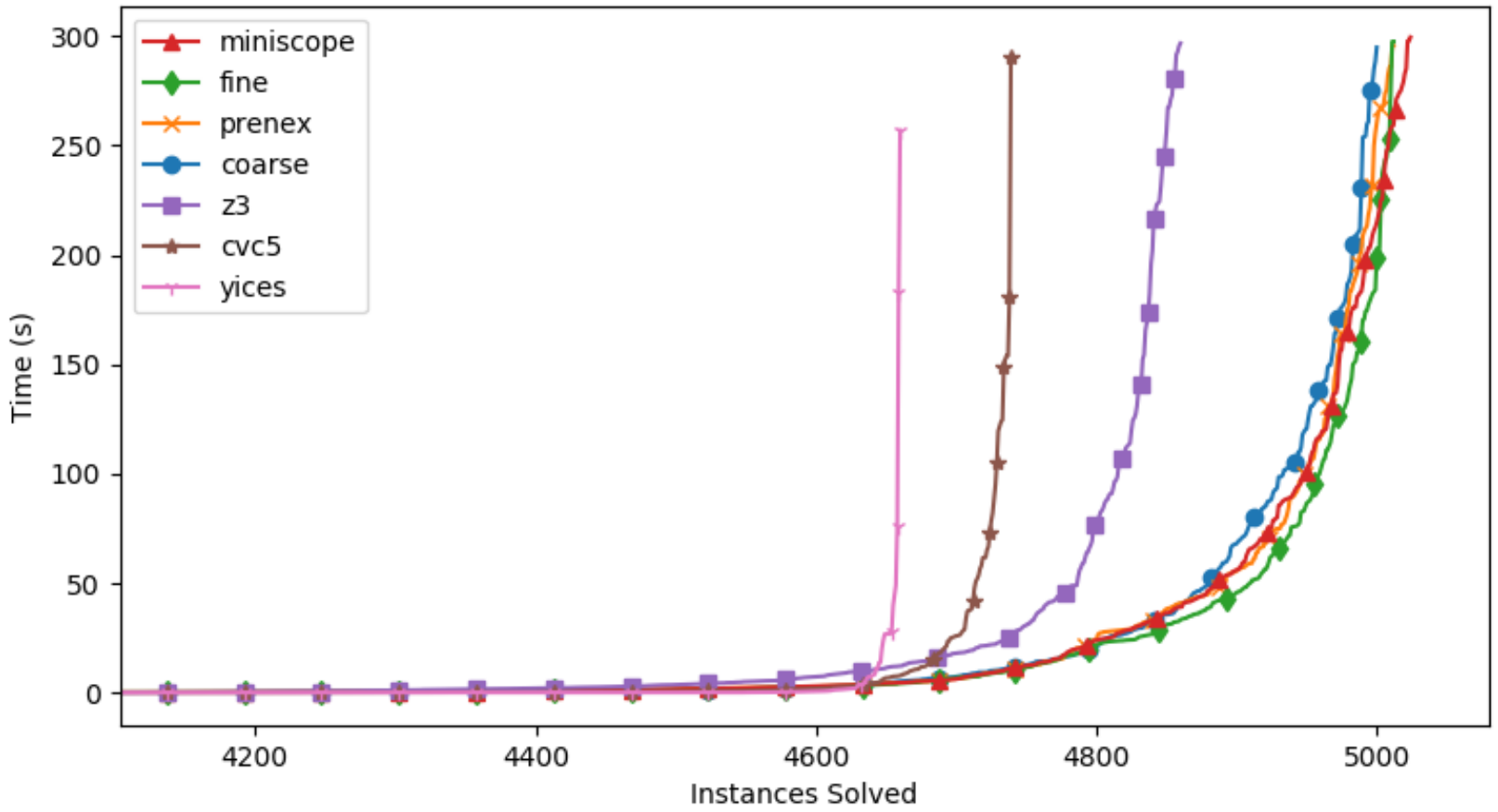
$lose_4(x,z) \stackrel{\text{def}}{=} z \leq x$

$lose_5(x,z) \stackrel{\text{def}}{=} z \leq x$

$lose_6(x,z) \stackrel{\text{def}}{=} z \leq x$

Evaluation

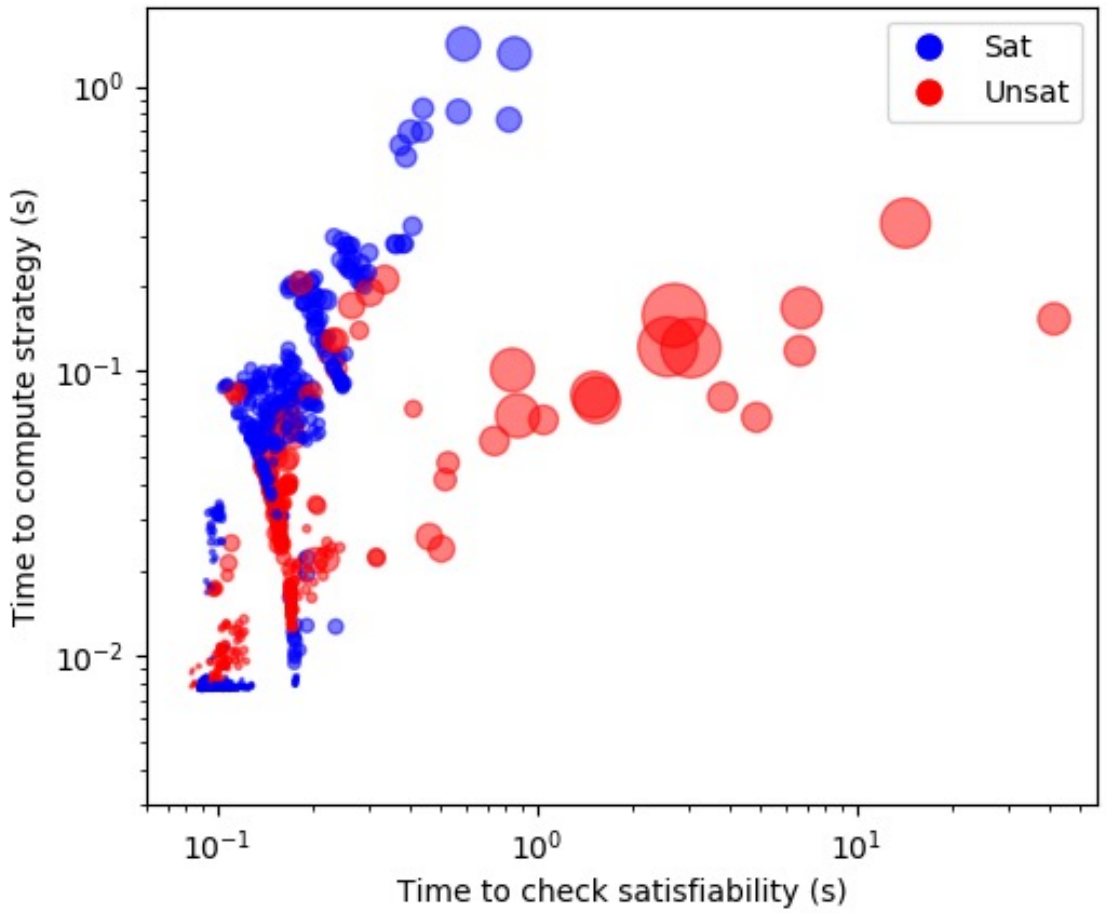
Evaluation


miniscope
fine
prenex
coarse
z3
cvc5
yices
Time (s)
300
250
200
150
100
50
0
4200
4400
4600
4800
5000
Instances Solved

Evaluation

Benchmarks	Miniscope	Fine	Prenex	Coarse	CVC5	YicesQS	Z3	Any	All	Total
Simulation	**2060**	**2060**	**2060**	2059	2059	1972	**2060**	2060	1972	2060
UltimateAutomizer	316	316	315	315	**345**	82	242	349	60	372
psyco	**189**	**189**	**189**	**189**	**189**	146	**189**	189	146	189
tptp (LIA)	**46**	**46**	**46**	**46**	**46**	42	**46**	46	42	46
Termination	**200**	**200**	**200**	196	195	0	166	200	0	200
All LIA	2811	2811	2810	2805	**2834**	2242	2703	2850	2220	2867
Mjollnir	1597	1584	1586	1578	1300	**1800**	1541	1800	1177	1800
keymaera	**222**	**222**	**222**	**222**	**222**	**222**	**222**	222	222	222
Scholl	372	373	372	373	362	**374**	372	374	359	374
tptp (LRA)	**23**	**23**	**23**	**23**	**23**	**23**	0	23	0	23
All LRA	2214	2202	2203	2196	1907	**2419**	2135	2419	1781	2419
All	**5025**	5013	5013	5001	4741	4661	4861	5269	4001	5286

Evaluation


Sat
Unsat
10^0
10^{-1}
10^{-2}
Time to compute strategy (s)
10^{-1}
10^0
10^1
Time to check satisfiability (s)

Questions

Thank you!

Questions

- Strategy Improvement:
  - Iteratively improve a strategy skeleton by recursively solving sub-games
  - Parametrized by $select_{\mathrm{T}}$ for a given theory
  - Sufficient for proving Satisfiability
- Strategy Synthesis:
  - Produces a winning strategy from a winning strategy skeleton
    - Can be further used for program synthesis/verification tasks