Inference in Cyc

- ***Logical Aspects of Inference***
- Incompleteness in Searching
- Incompleteness from Resource Bounds and Continuable Searches
- Inference Features in Cyc
- Efficiency through Heuristics

CYCORP

Inference uses Deduction: Rules

“Rules” - general, variables

```
(#$implies
 (#$mother ?PERSON ?MOTHER)
 (#$loves  ?PERSON ?MOTHER))
```

Inference uses Deduction: Facts

“Facts” - specific, no variables, atomic

(#$mother #$Hamlet #$Gertrude)

CYCORP

Inference uses Deduction: Non-atomic terms, Predicates, and Functions

“Non-atomic” terms are functional

(#$BabyFn #$Jaguar)

Predicates are true or false

(#$mother #$Hamlet #$Gertrude)

Functions denote a new term

(#$BabyFn #$Jaguar)

CYCORP

Inference uses Deduction: Formulas and Logic

“Formula” - a relation applied to arguments

```
(#$implies
 (#$mother ?PERSON ?MOTHER)
 (#$loves  ?PERSON ?MOTHER))
```

Cyc’s Inference uses standard logical deductions

All men are mortal.

Socrates is a man.

Socrates is mortal.

CYCORP

Inference uses Deduction: Rules + Facts

Deduction -

rule + fact(s) => new fact

```
(#$loves #$Hamlet #$Gertrude)
```

“Rules” - general, variables

```
(#$implies
 (#$mother ?PERSON ?MOTHER)
 (#$loves  ?PERSON ?MOTHER))
```

“Facts” - specific, no variables

```
(#$mother #$Hamlet #$Gertrude)
```

The Resolution Principle

Resolution Principle : “Unify, Substitute, Merge”

Query

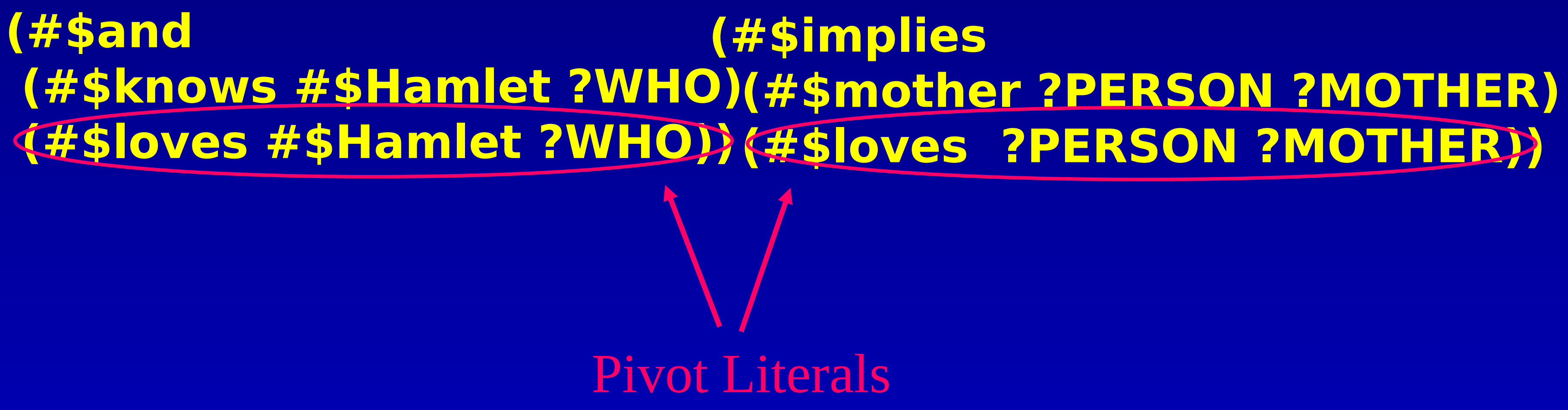
```
(#$and
 (#$knows #$Hamlet ?WHO)
 (#$loves #$Hamlet ?WHO))
```

Rule

```
(#$implies
(#$mother ?PERSON ?MOTHER)
(#$loves  ?PERSON ?MOTHER))
```

The Resolution Principle: Unify

Resolution Principle : “***Unify***, Substitute, Merge”



Query

```
(#$and
 (#$knows #$Hamlet ?WHO)
 (#$loves #$Hamlet ?WHO))
```

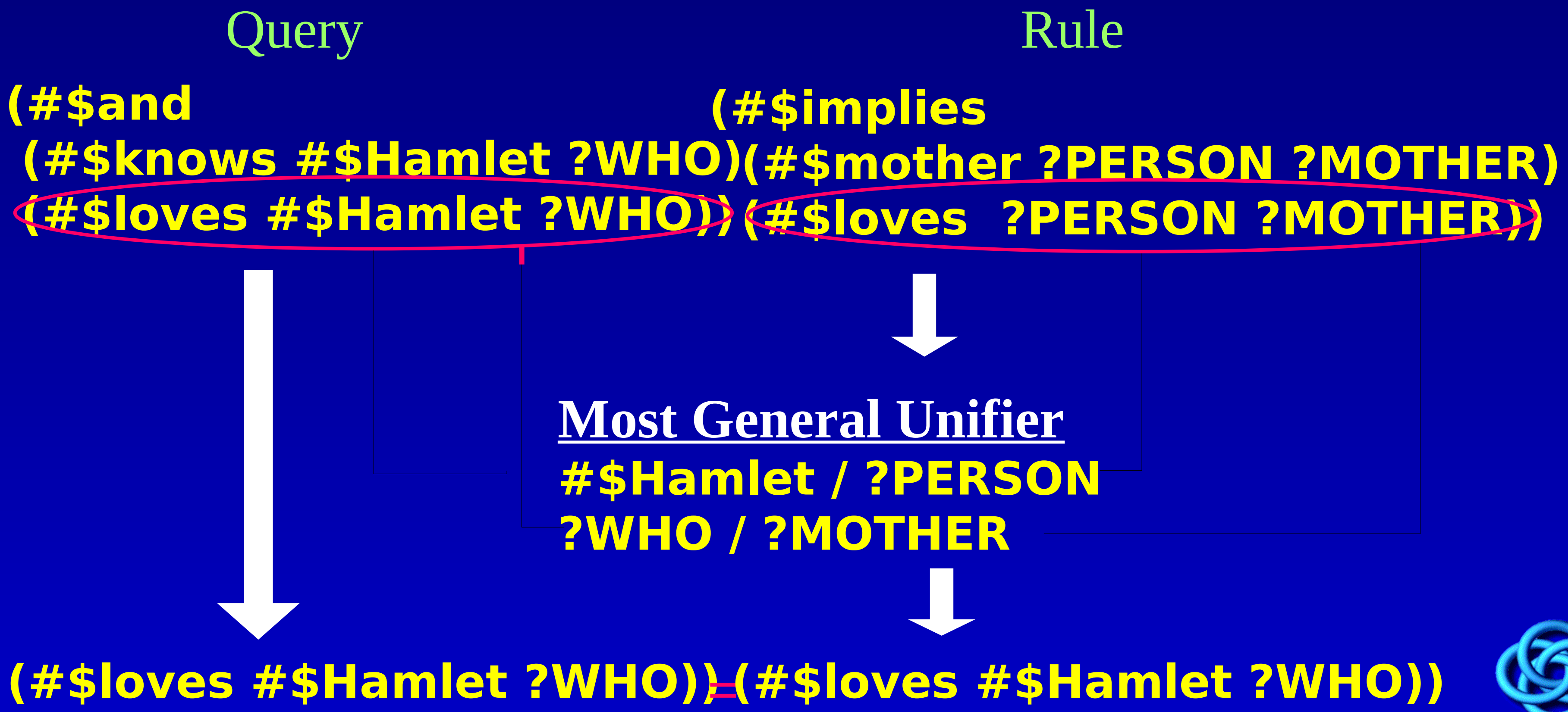
Rule

```
(#$implies
(#$mother ?PERSON ?MOTHER)
(#$loves  ?PERSON ?MOTHER))
```

Pivot Literals

The Resolution Principle: Unify

Resolution Principle : "***Unify***, Substitute, Merge"



Query

(#$and
(#$knows #$Hamlet ?WHO)
(#$loves #$Hamlet ?WHO))

Rule

(#$implies
(#$mother ?PERSON ?MOTHER)
(#$loves ?PERSON ?MOTHER))

<u>**Most General Unifier**</u>

#$Hamlet / ?PERSON
?WHO / ?MOTHER

(#$loves #$Hamlet ?WHO)) = (#$loves #$Hamlet ?WHO))

The Resolution Principle: Substitute

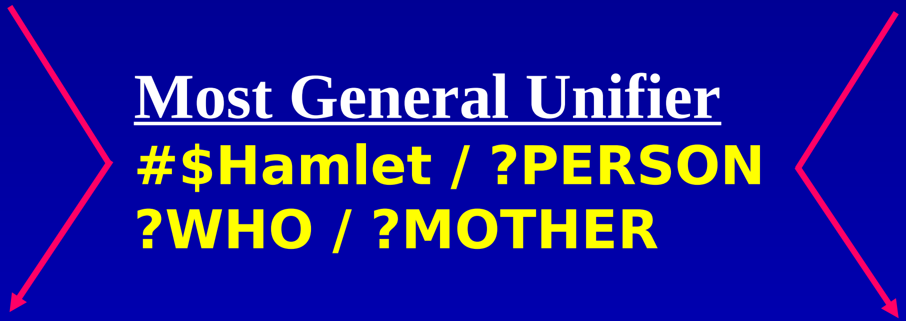
Resolution Principle : “Unify, ***Substitute***, Merge”

Query

```
(#$and
 (#$knows #$Hamlet ?WHO)
 (#$loves #$Hamlet ?WHO))
```

Rule

```
(#$implies
(#$mother ?PERSON ?MOTHER)
(#$loves  ?PERSON ?MOTHER))
```



Most General Unifier

#$Hamlet / ?PERSON
?WHO / ?MOTHER

Substituted Query

```
(#$and
 (#$knows #$Hamlet ?WHO)
 (#$loves #$Hamlet ?WHO))
```

Substituted Rule

```
(#$implies
(#$mother #$Hamlet ?WHO)
(#$loves #$Hamlet ?WHO))
```

The Resolution Principle: Merge

Resolution Principle : “Unify, Substitute, ***Merge***”

Substituted Query

```
(#$and
 (#$knows #$Hamlet ?WHO)
 (#$loves #$Hamlet ?WHO))
```

Substituted Rule

```
(#$implies
 (#$mother #$Hamlet ?WHO)
 (#$loves  ?Hamlet ?WHO))
```

Merged Query

```
(#$and
 (#$knows  #$Hamlet ?WHO)
 (#$mother #$Hamlet ?WHO))
```

CYCORP

The Resolution Principle using a fact

Resolution Principle : “Unify, Substitute, Merge”

Query

```
(#$and
 (#$knows #$Hamlet ?WHO)
 (#$loves #$Hamlet ?WHO))
```

Fact

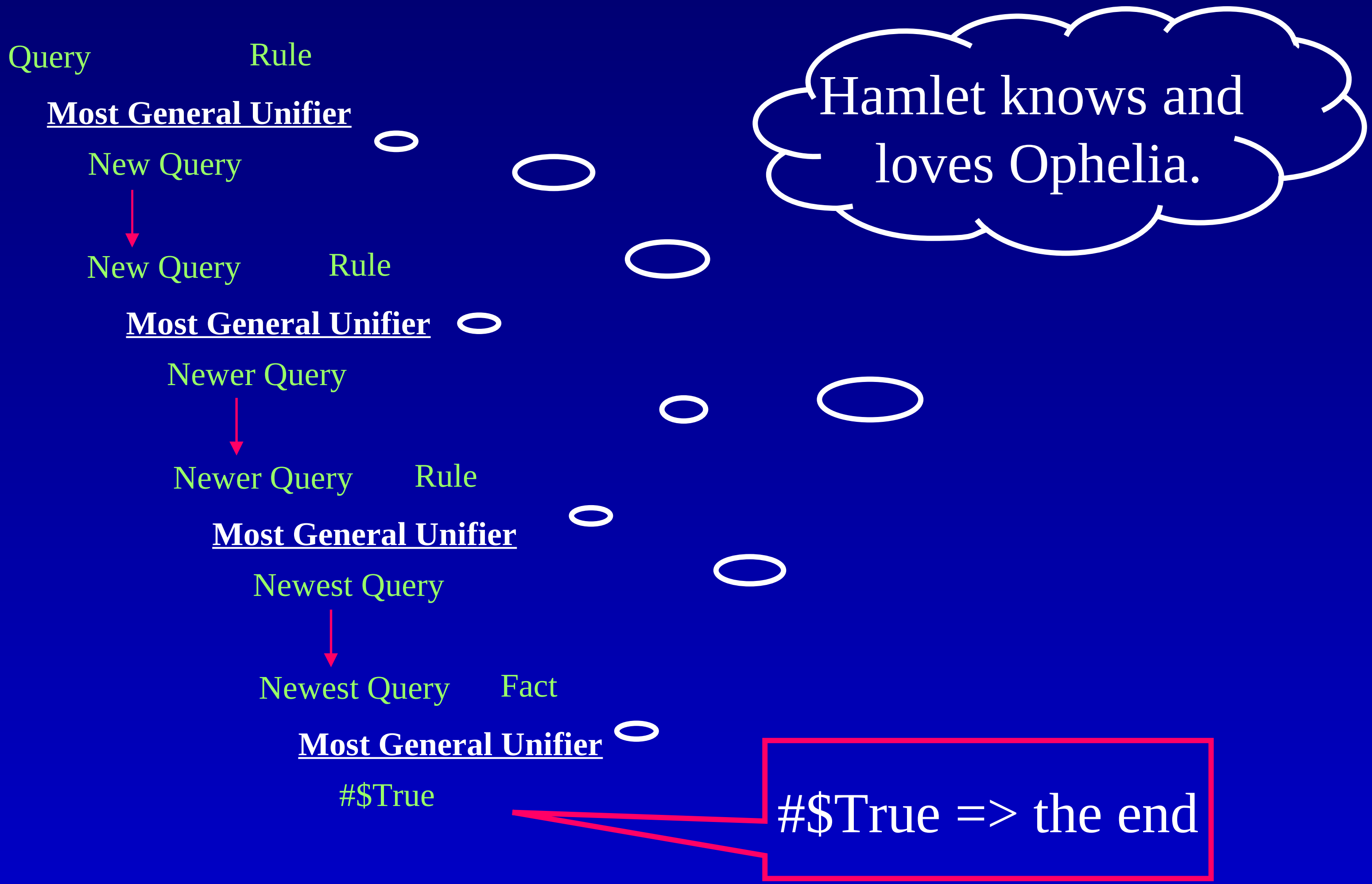
```
(#$loves #$Hamlet #$Ophelia)
```

Most General Unifier

?WHO / #$Ophelia

```
(#$knows  #$Hamlet #$Ophelia)
```

Resolving to #$True



Query Rule

Most General Unifier

New Query

New Query Rule

Most General Unifier

Newer Query

Newer Query Rule

Most General Unifier

Newest Query

Newest Query Fact

Most General Unifier

#$True

Hamlet knows and loves Ophelia.

#$True => the end

Summary

- Inference uses Deduction
  - Facts + Rules => New Fact
  - Rules vs. Facts
  - Predicates vs. Functions
- Inference uses Resolution
  - The Resolution Principle: Unify, Substitute, Merge
  - Resolving to #$True

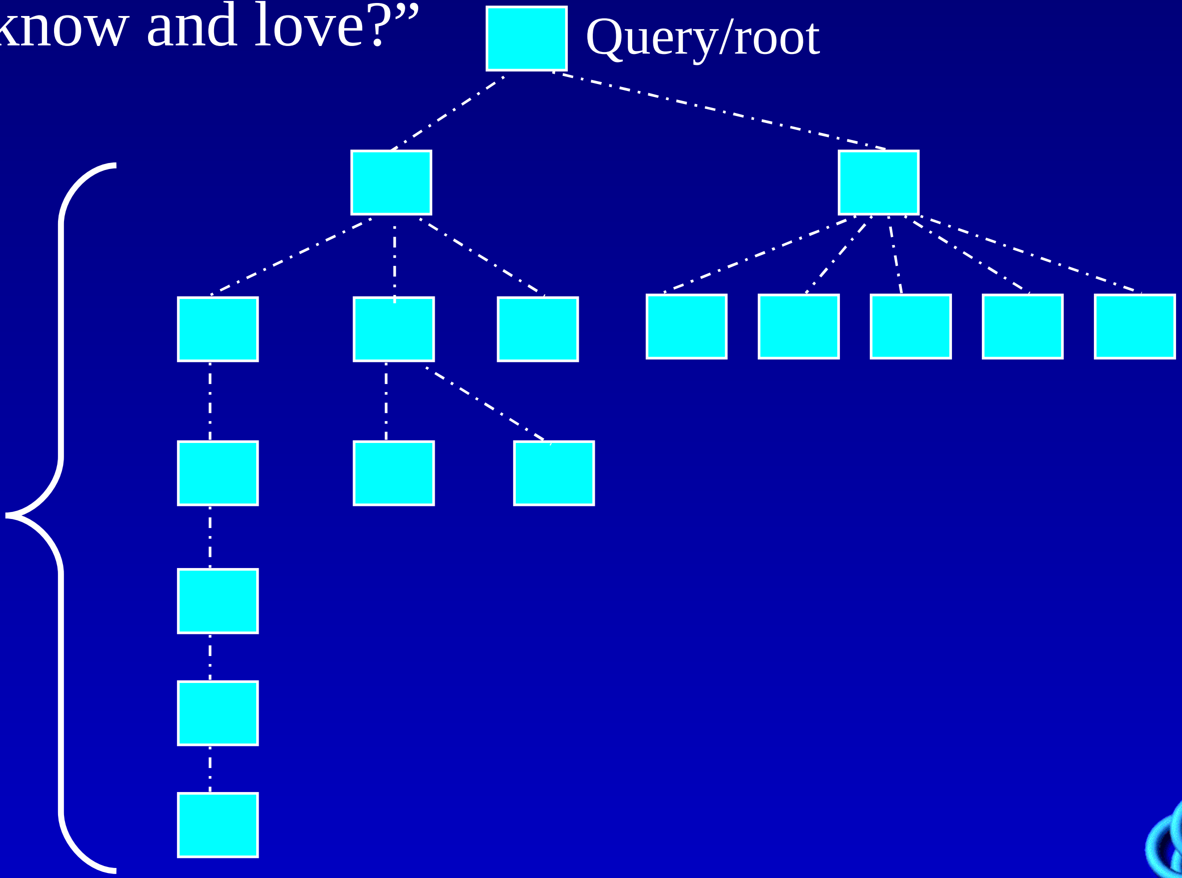
CYCORP

Inference in Cyc

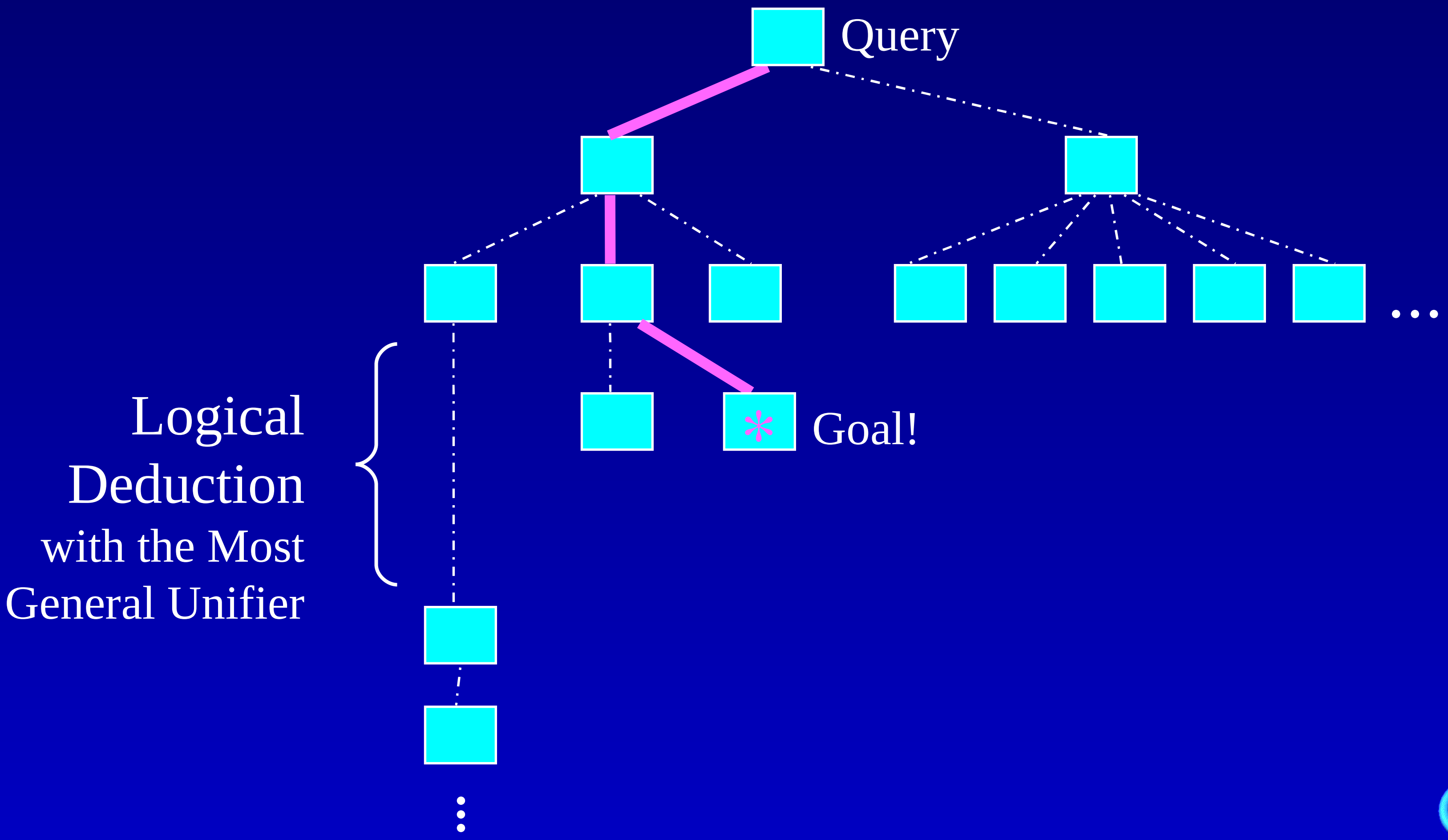
- Logical Aspects of Inference
- ***Incompleteness in Searching***
- Incompleteness from Resource Bounds and Continuable Searches
- Inference Features in Cyc
- Efficiency through Heuristics

The Search Tree

“Who does Hamlet know and love?”



Justifying the Answers



CYCORP

Justifying the Answers (cont.'d)

- Goal Node
  - "true"
  - "yes, I was able to prove it true"

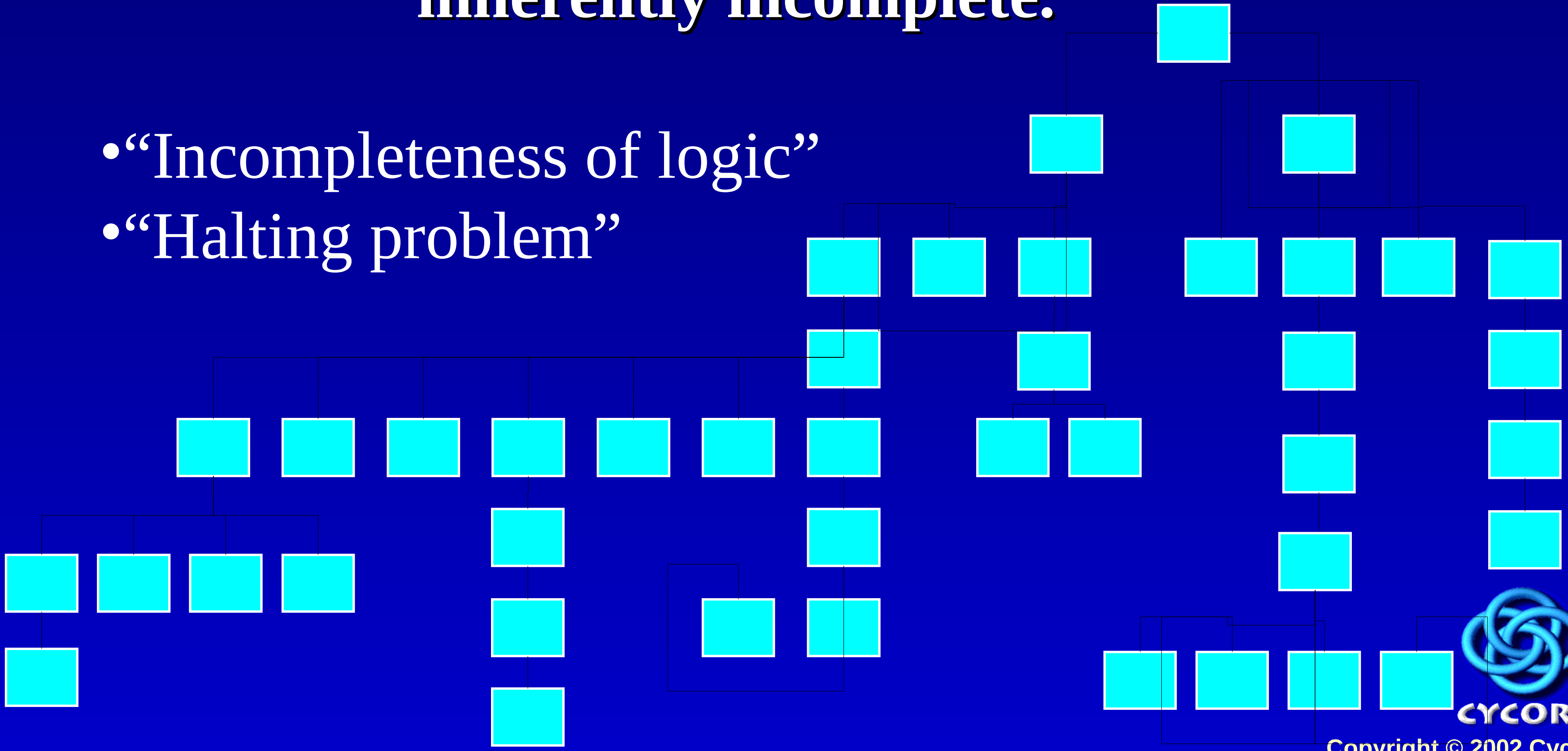
- Bindings
  - values assigned to variables in the unification process
  - only includes values that were used in successful proofs
  - included in the justification

CYCORP

Incompleteness in Searching

Any sufficiently complicated logical system is inherently incomplete.

- “Incompleteness of logic”
- “Halting problem”

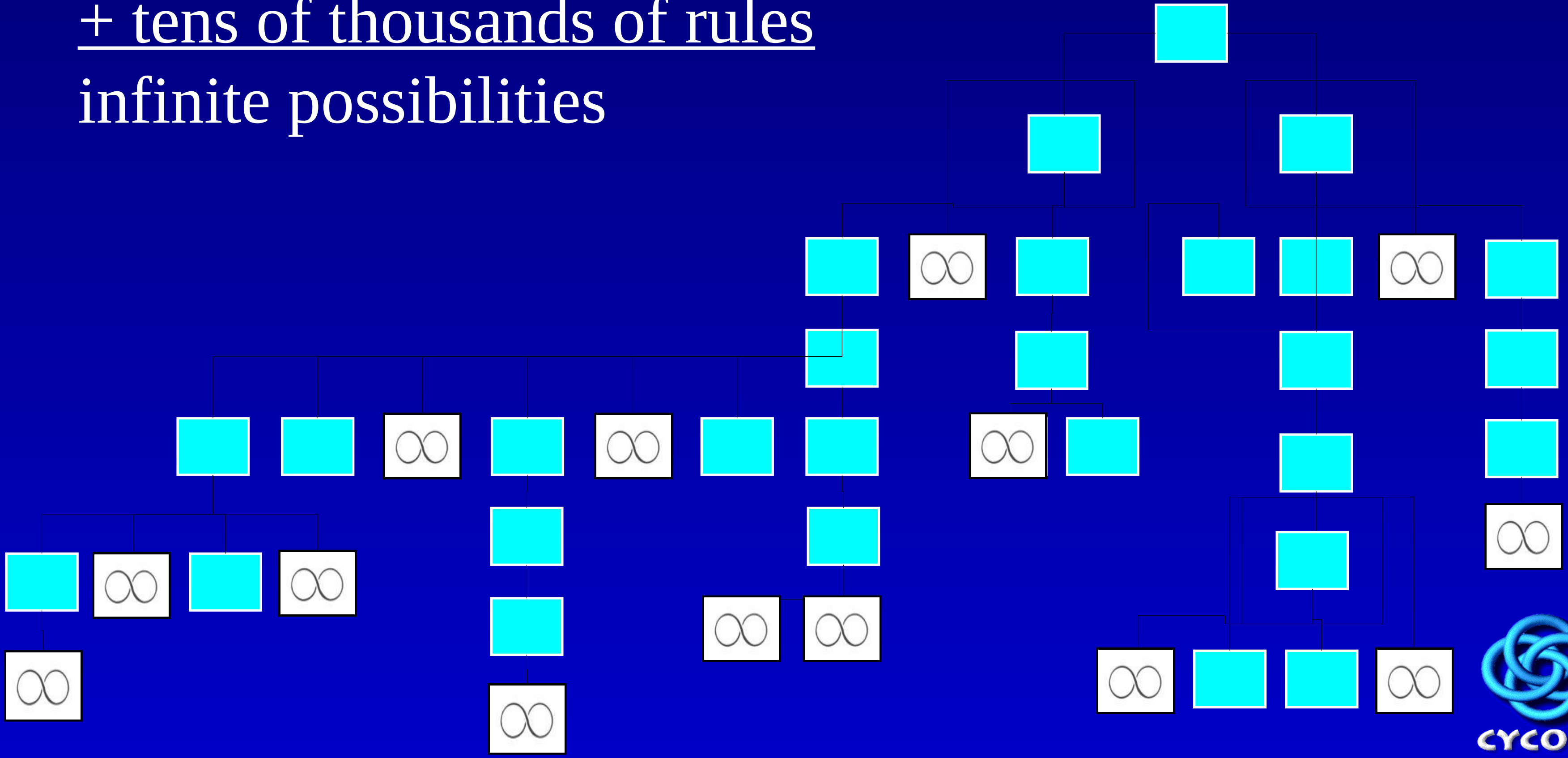


The Halting Problem

millions of facts

+ tens of thousands of rules

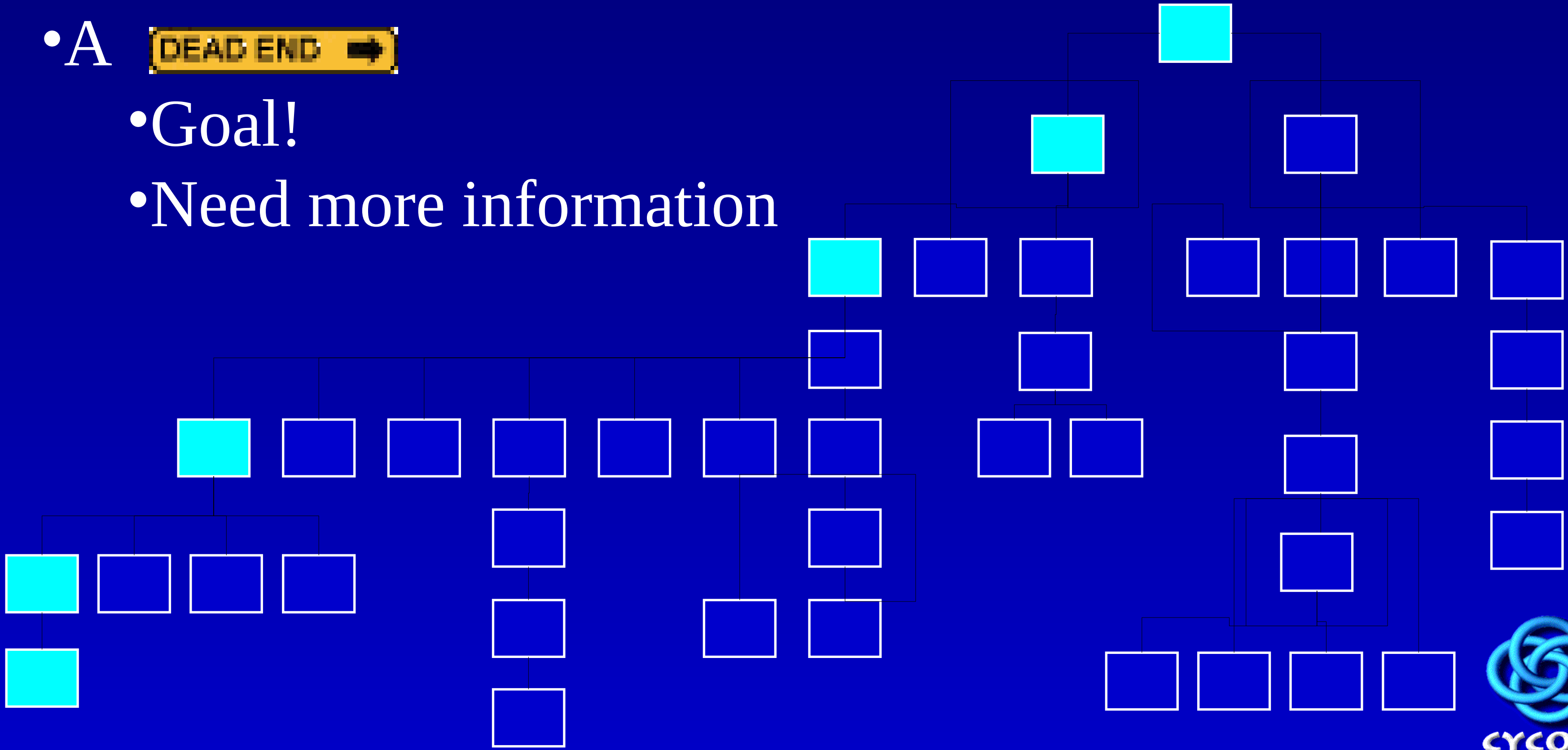
infinite possibilities



CYCORP

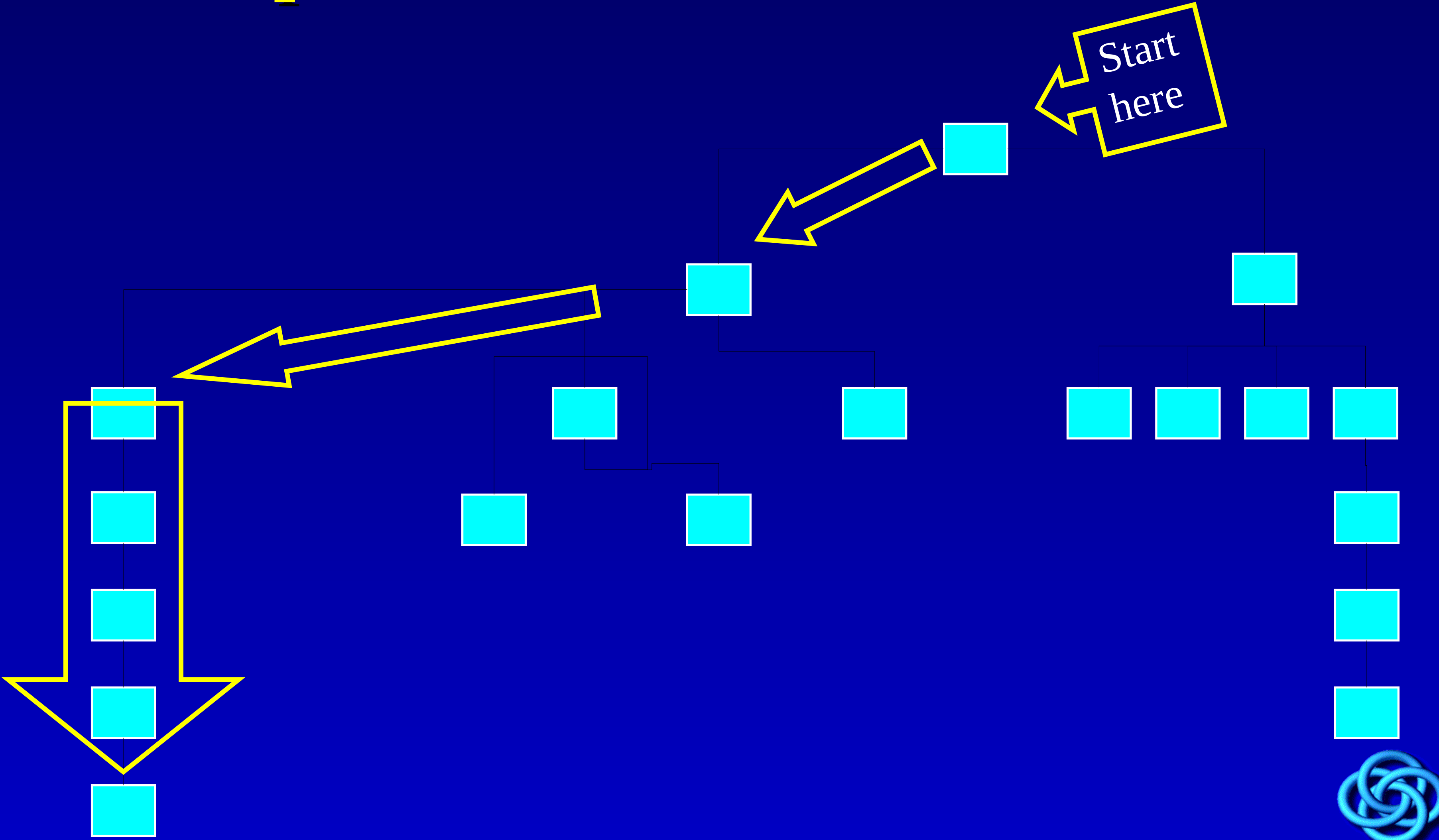
Depth-first Search

- Start at the top
- Expand as you go down
- A DEAD END ➡
  - Goal!
  - Need more information

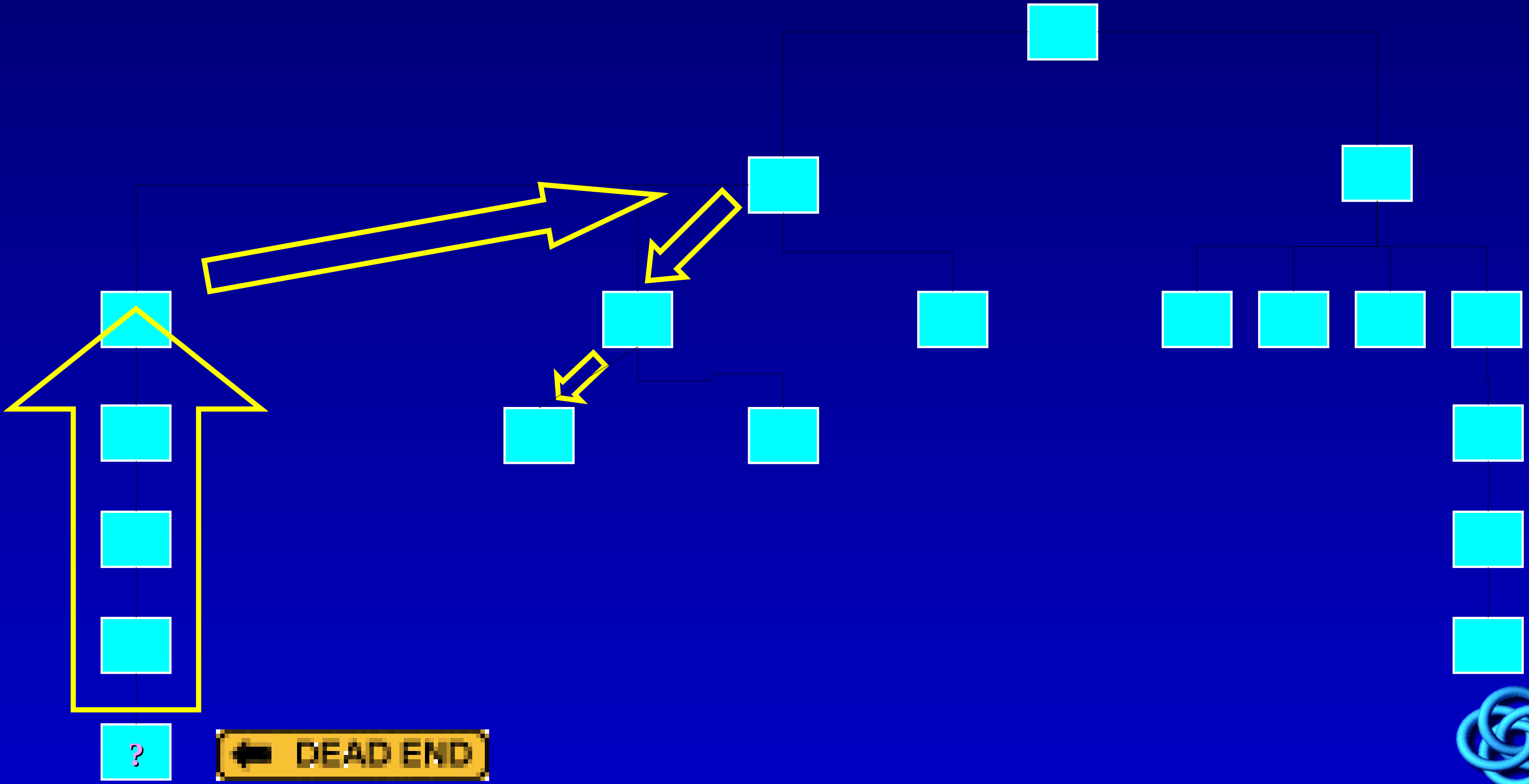


CYCORP

Depth-first Search Traversal

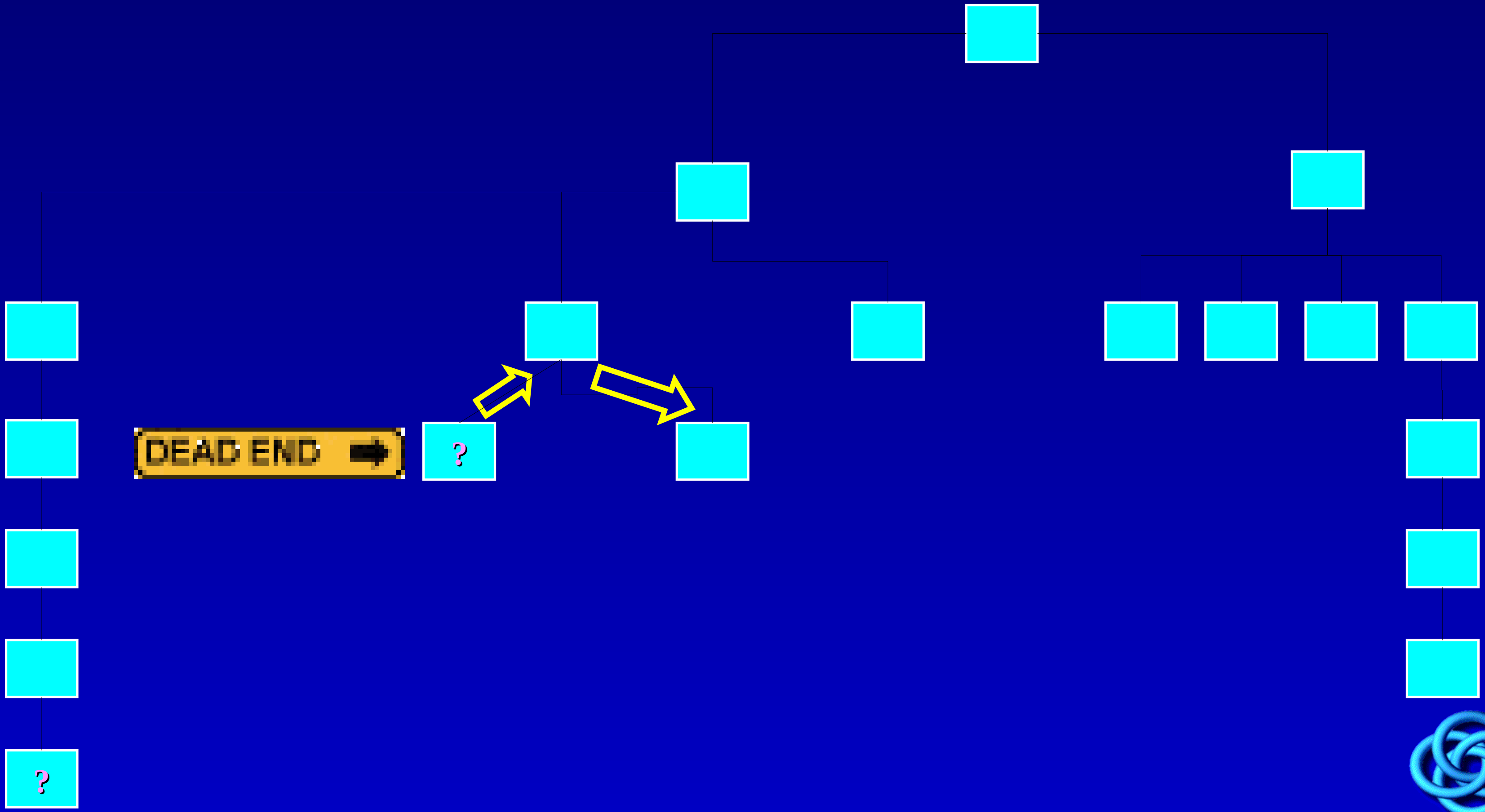


Depth-first Search Traversal

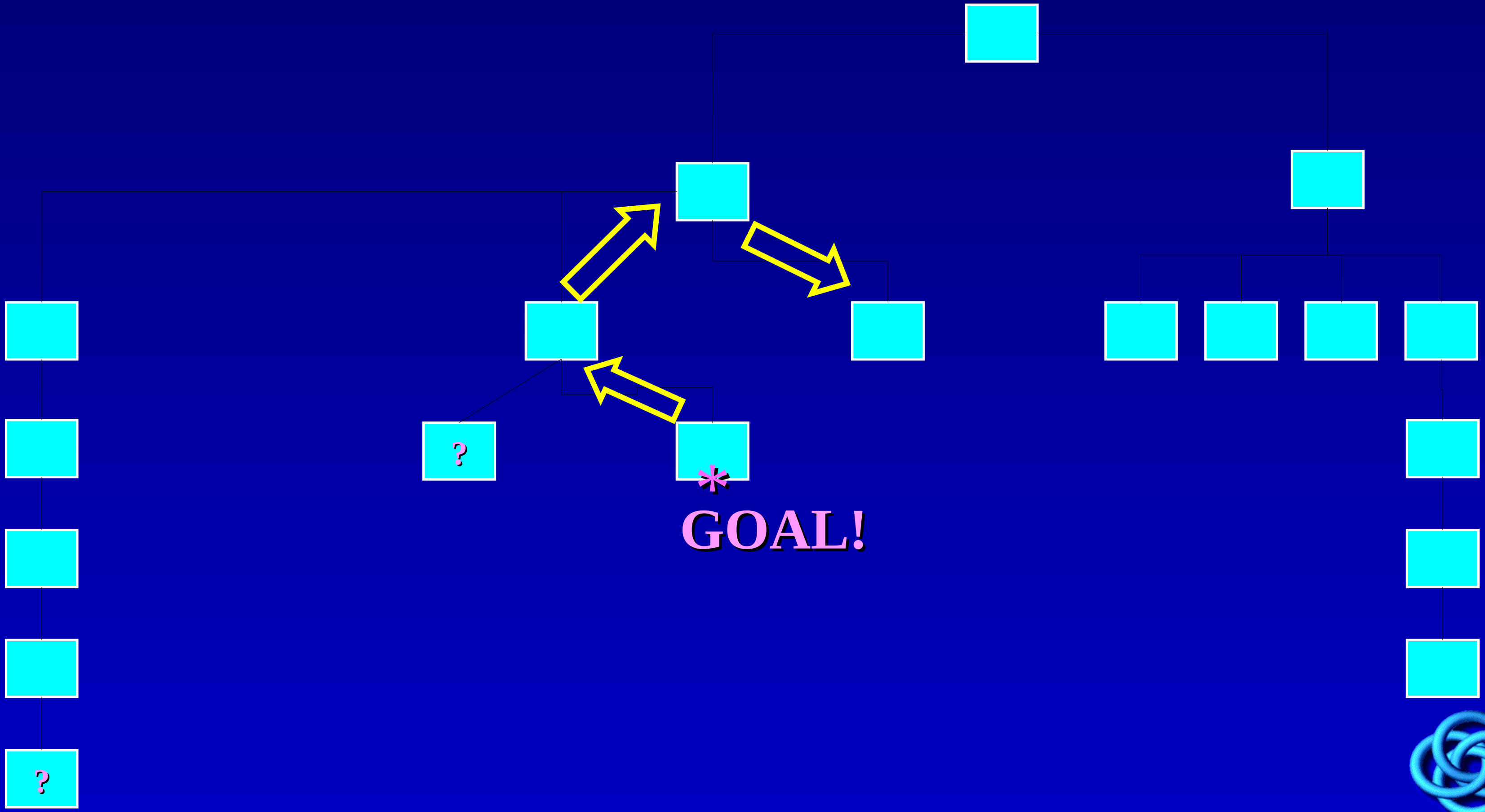


Copyright © 2002 Cycorp

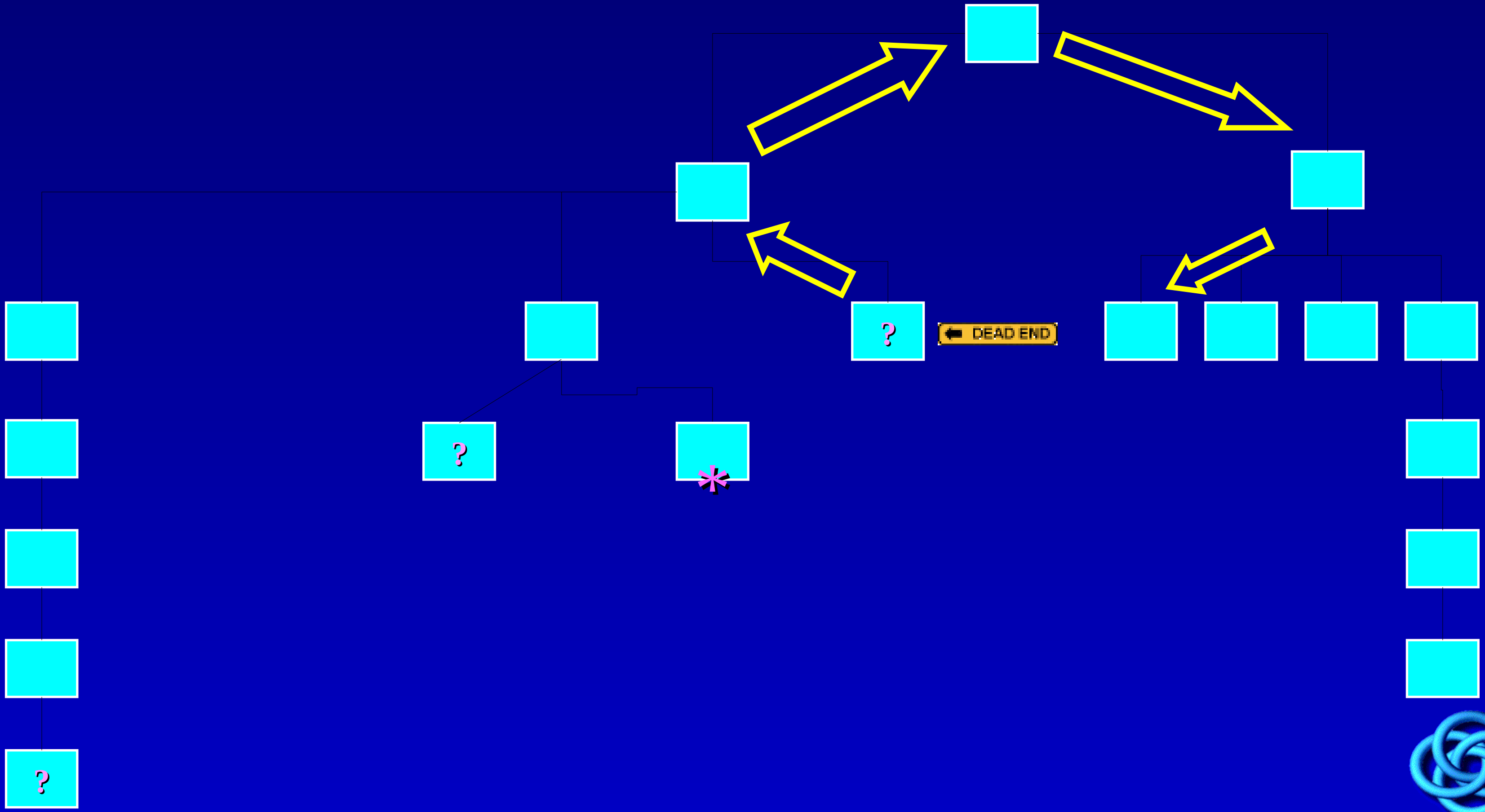
Depth-first Search Traversal



Depth-first Search Traversal

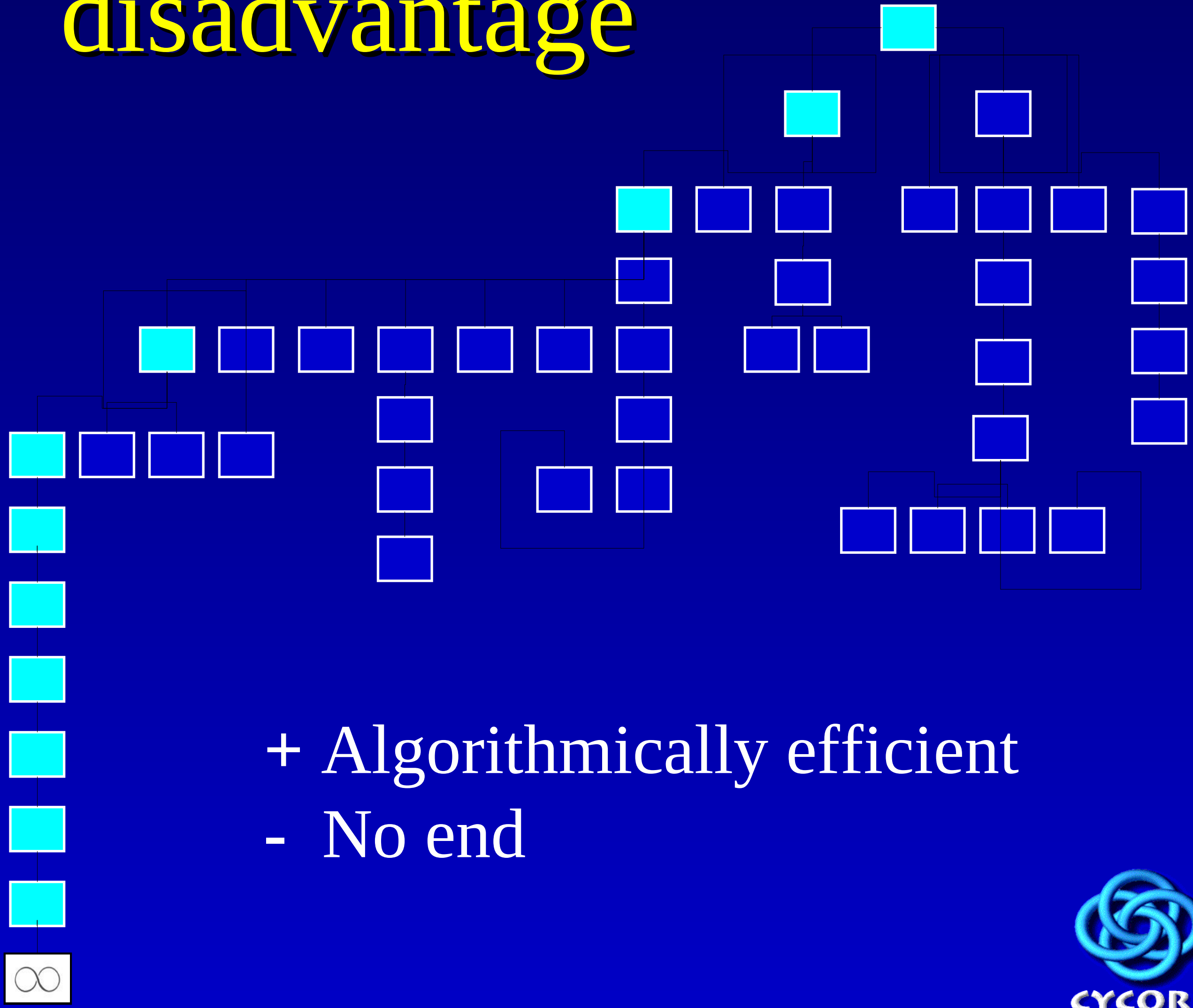


Depth-first Search Traversal



Copyright © 2002 Cycorp

Depth-first: advantage and disadvantage

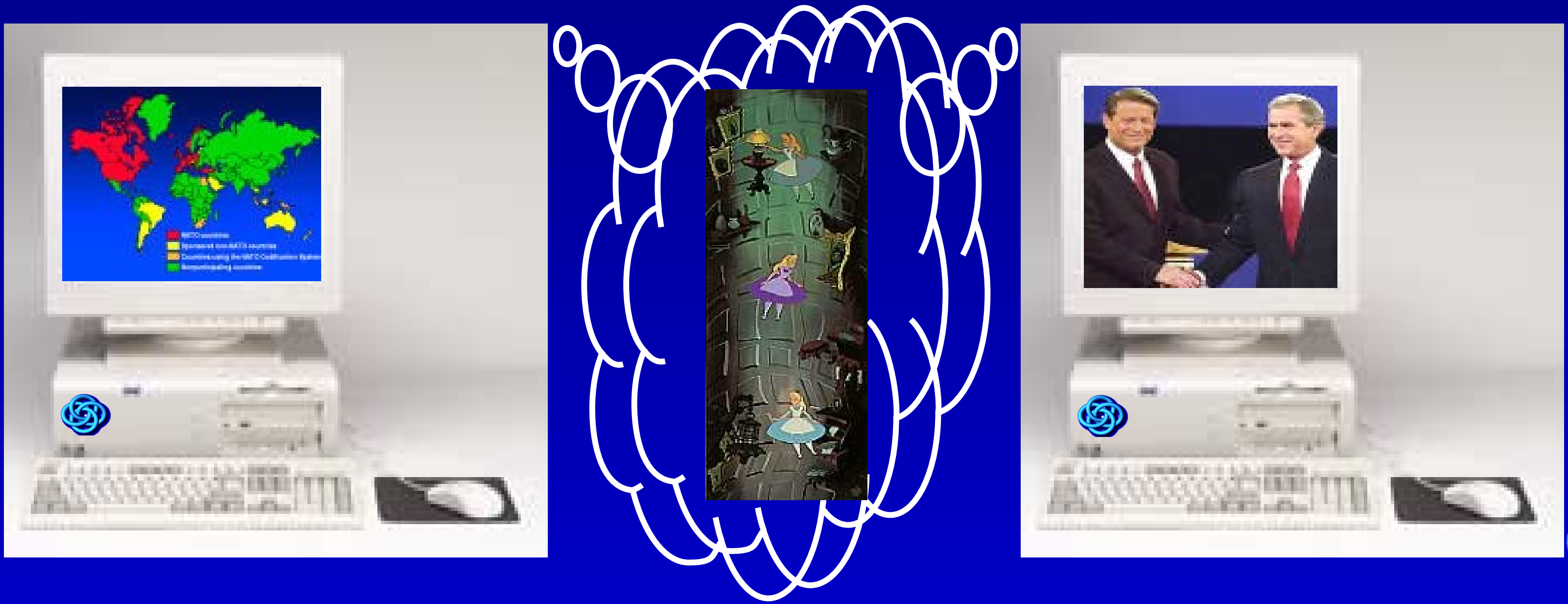


+ Algorithmically efficient

- No end

A Depth-first Rabbit Hole: “Cyc-ic Friends”

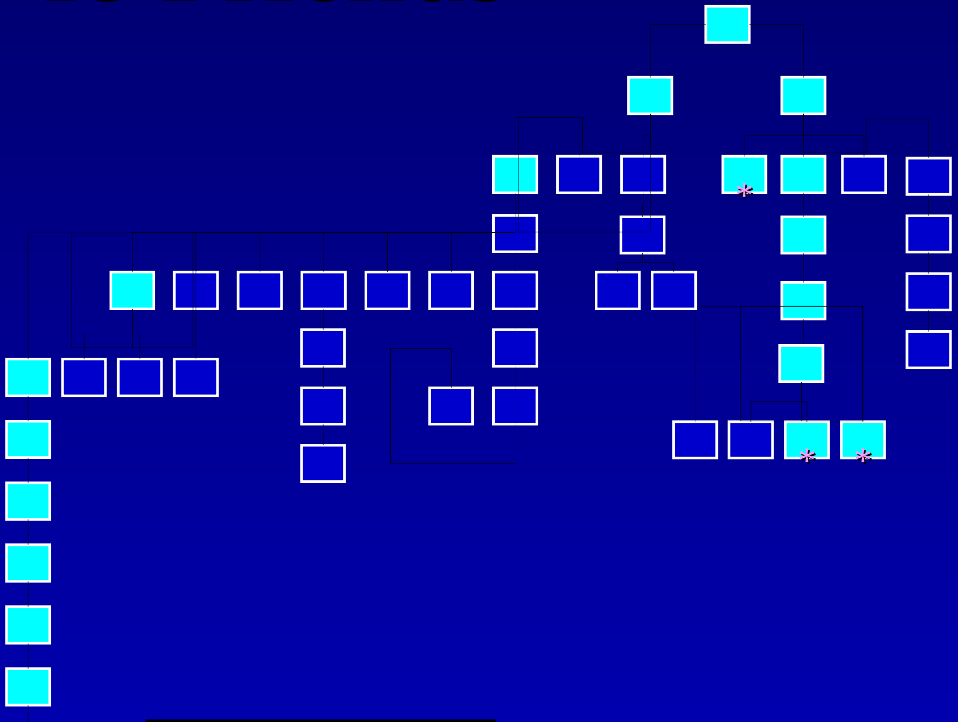
“Who are all of the elected leaders of countries north of the equator?”

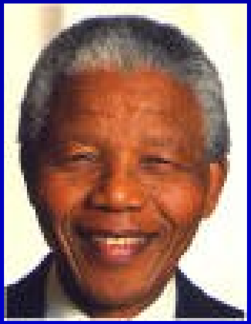


CYCORP

A Depth-first Rabbit Hole: "Cyc-ic Friends"

The Halting Problem caused Cyc to keep trying, even when the logic (to us) was completely nonsensical….



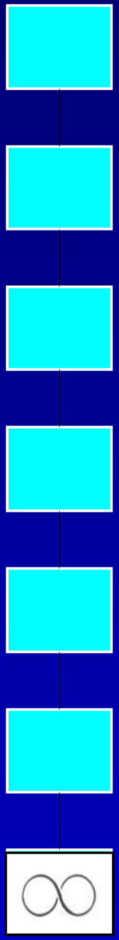


=

?

CYCORP

A Depth-first Rabbit Hole: “Cyc-ic Friends”



How did it get so far off?

- Steps were logical
- Steps involved default-true rules
  - usually true in isolation
  - need more logic when so many are chained together

The Halting Problem: a Trade-off

The Halting Problem:

- Should I just try and do a little more work to try and prove this?
- Can I prove that I shouldn't even be bothering to try and do this proof?

Maybe you're missing a key constraint….

CYCORP

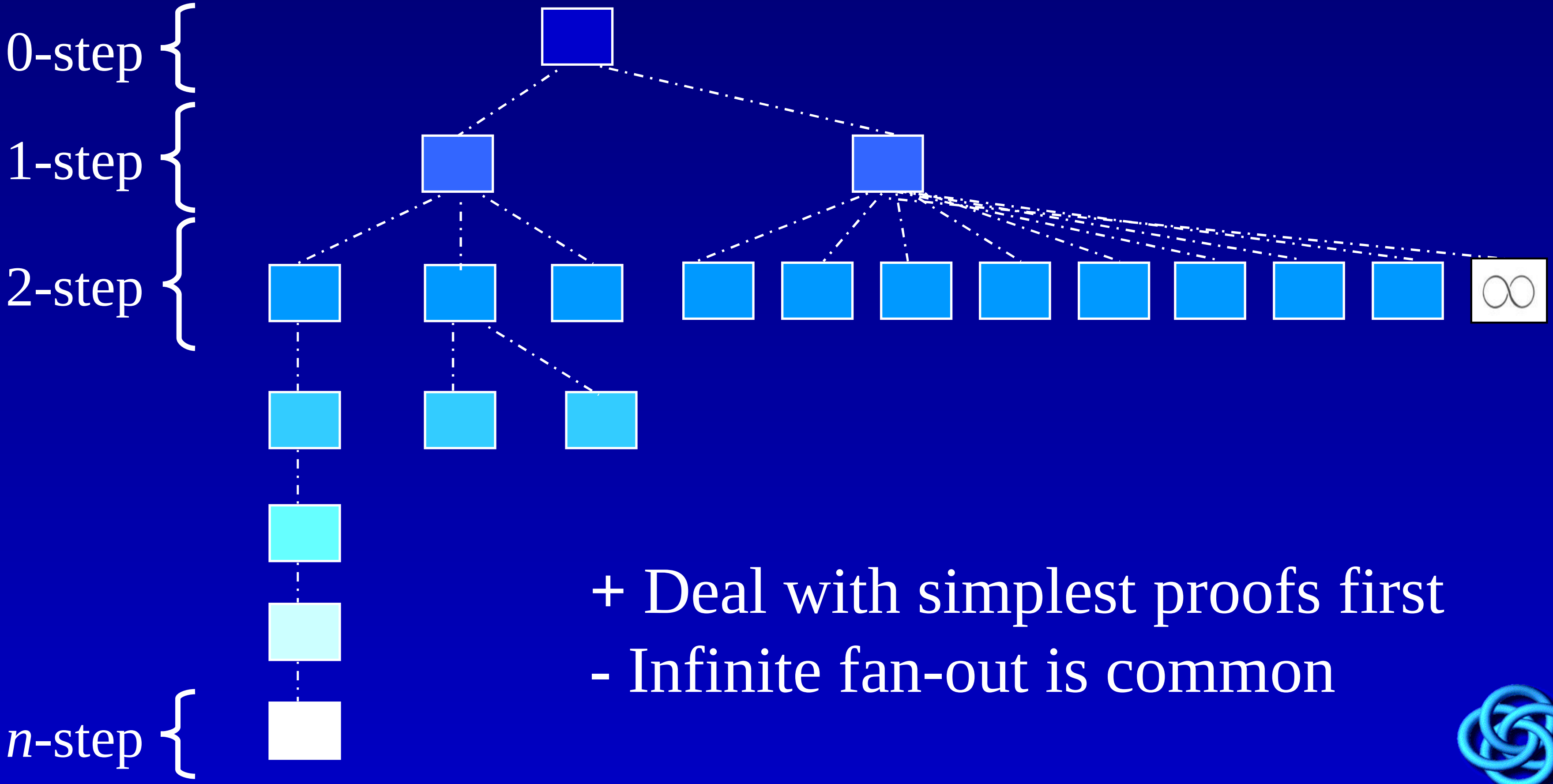
The Halting Problem: a Trade-off

Trying to prove that I shouldn’t even be bothering to try and do this proof….

- Proof by contradiction
- Can I prove that this couldn’t possibly be true?

Work harder or work smarter?

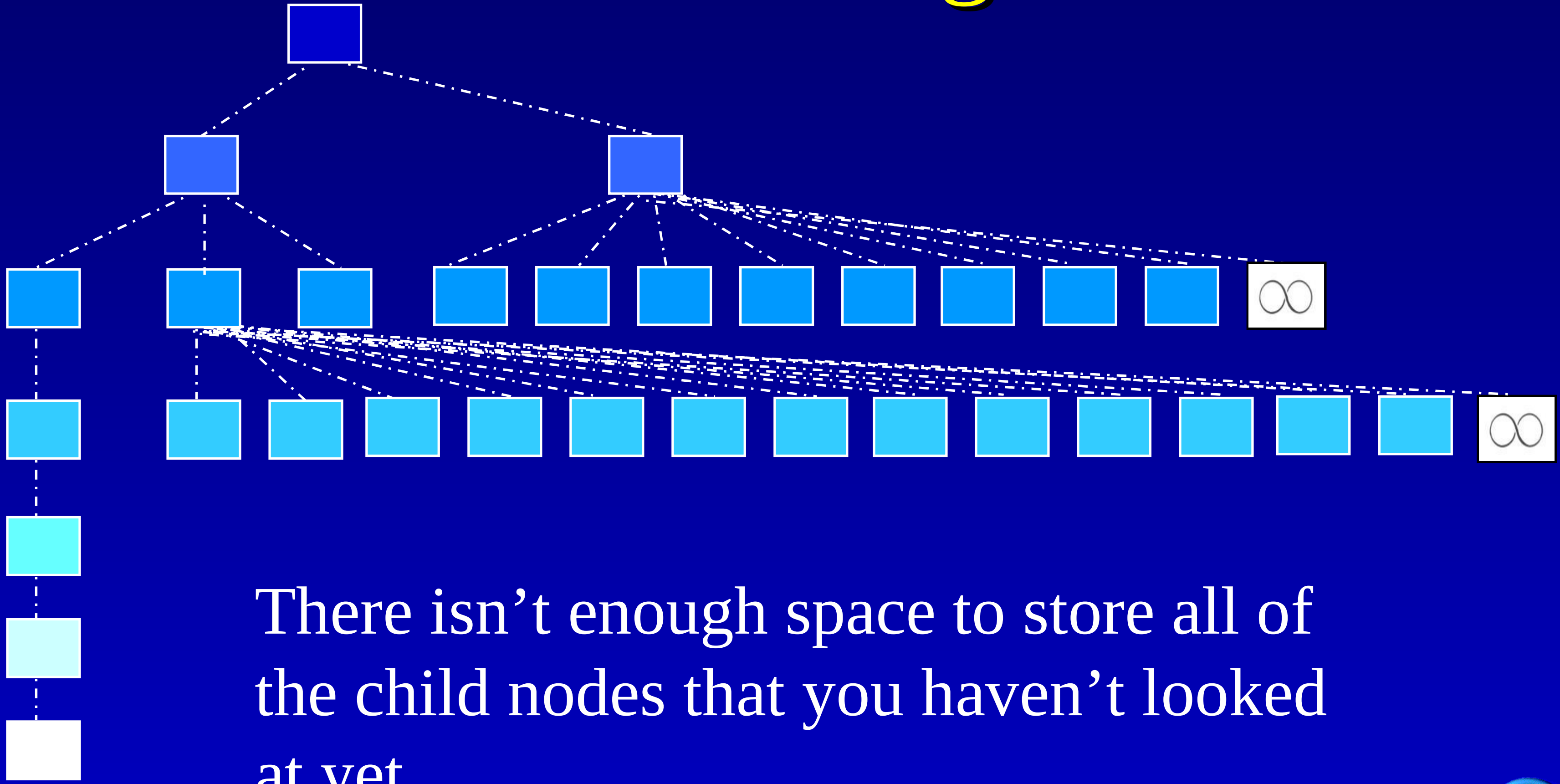
Breadth-first Search: advantage and disadvantage



+ Deal with simplest proofs first

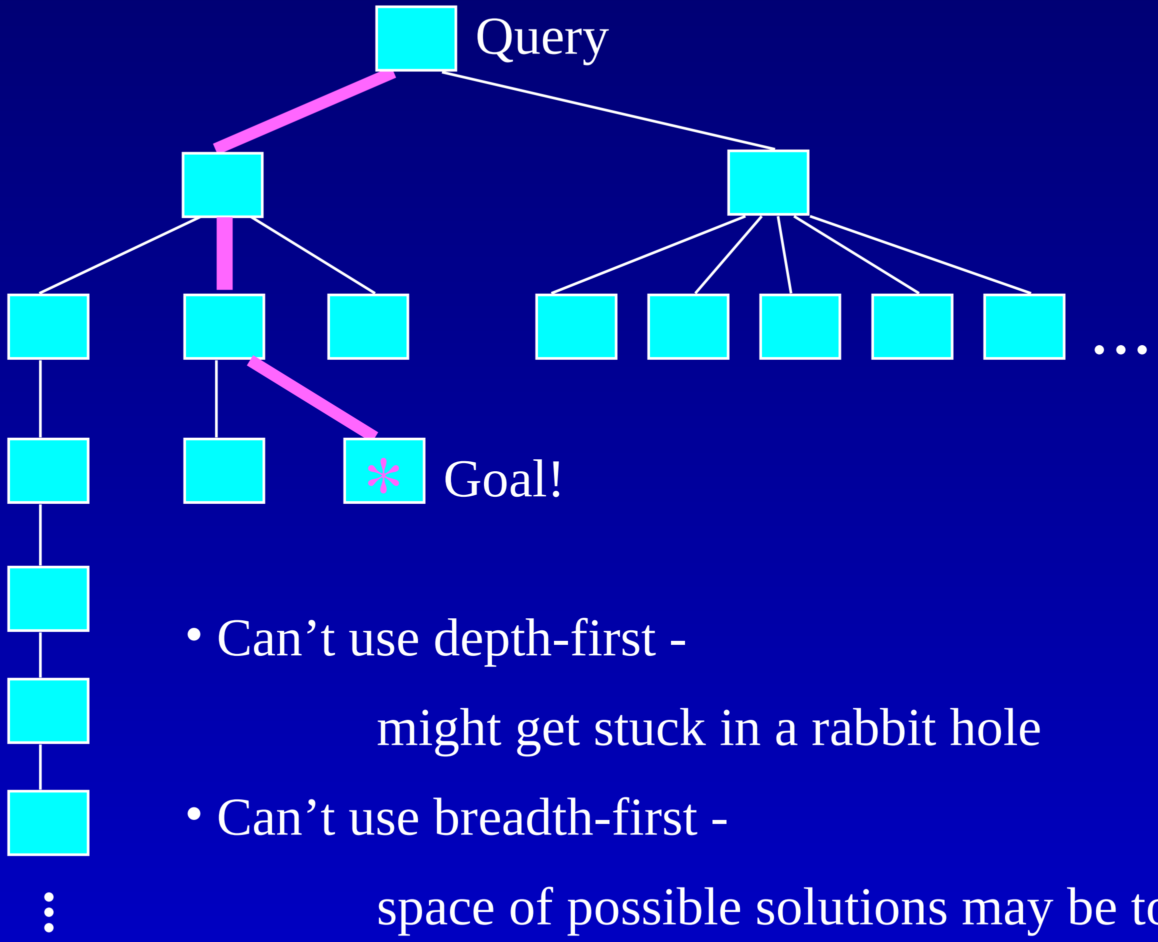
- Infinite fan-out is common

Breadth-first Search: another disadvantage



There isn’t enough space to store all of the child nodes that you haven’t looked at yet.

We Need a Better Search



- Can’t use depth-first -

  might get stuck in a rabbit hole

- Can’t use breadth-first -

  space of possible solutions may be too big

Summary

- The Search Tree
- Incompleteness of Logic
- The Halting Problem
  - Infinite Possibilities
  - Work harder or work smarter?
- Depth-first Searches
  - Rabbit-holes
- Breadth-first Searches
  - Infinite fan-out
  - Infinite space required to store possible solutions

Inference in Cyc

- Logical Aspects of Inference
- Incompleteness in Searching
- ***Incompleteness from Resource Bounds and Continuable Searches***
- Inference Features in Cyc
- Efficiency through Heuristics

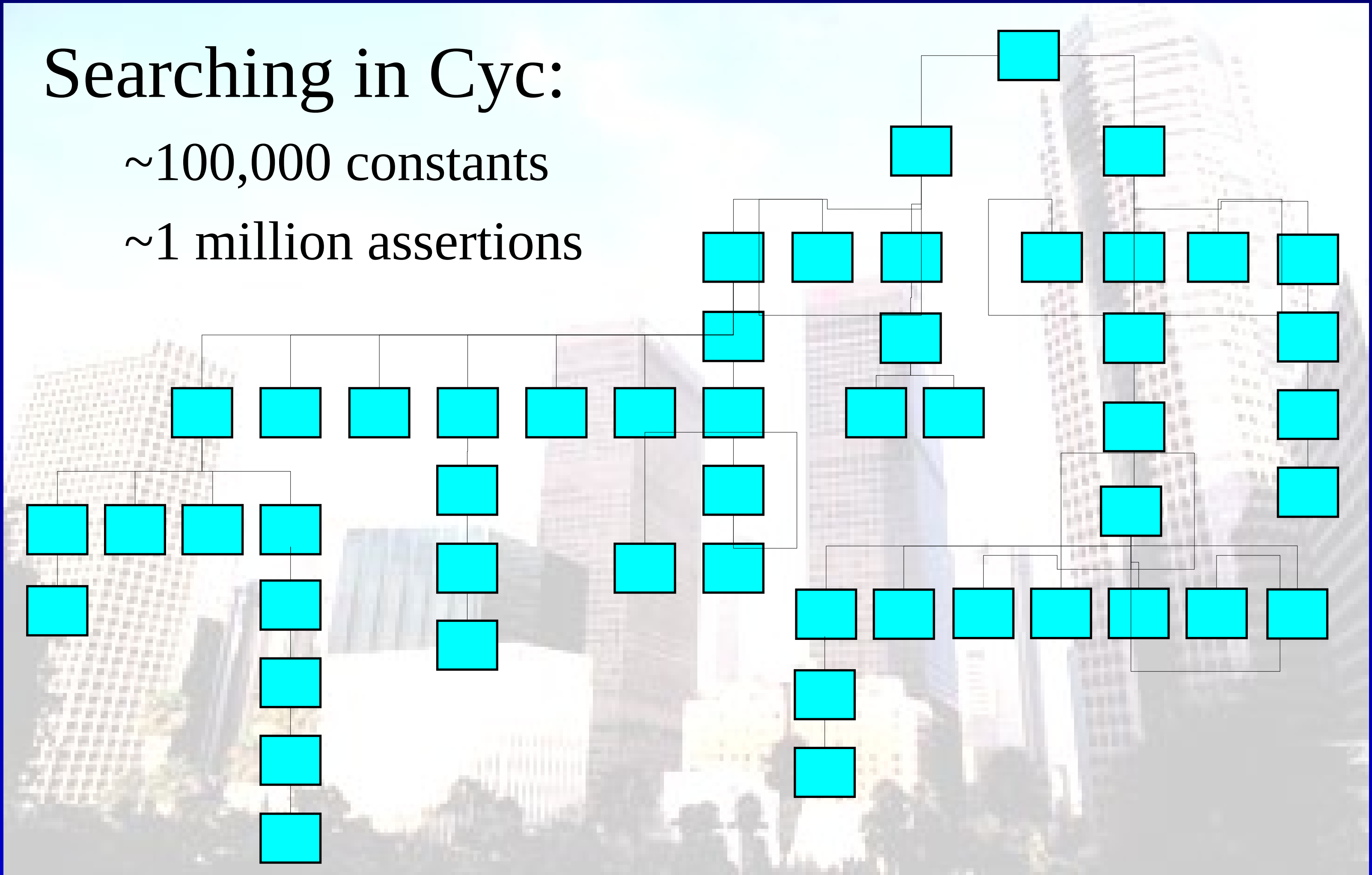
CYCORP

Cyc is Life in the Big City

Searching in Cyc:

~100,000 constants

~1 million assertions



Inference is Resource-bounded

Resource Bounds :

- quit after NUMBER of answers
- quit after TIME seconds
- ignore any proof using more than BACKCHAIN rules
- ignore any proof using more than DEPTH steps

CYCORP

Resource-bounded Incompleteness

In order to deal with life in the big city:

- The algorithm decides how completely to search
- The user sets the resource bounds for the search

or else the computer runs out of memory….

Inference is Continuable

If you quit early, you need to be able to pick up where you left off:

- Proof search state is explicitly maintained
- Can continue with additional resources

Proof Search Could be Stored With Meta Data

Proof search state could be explicitly maintained along with bookkeeping data to make future searches efficient

Proof Search Could be Stored With Meta Data (cont.'d)

Meta data could be used in heuristic searches

- a path might look promising according to past experience with
  - certain proofs
  - certain rules
  - rules chained together in a certain way
- store this data explicitly with the search tree

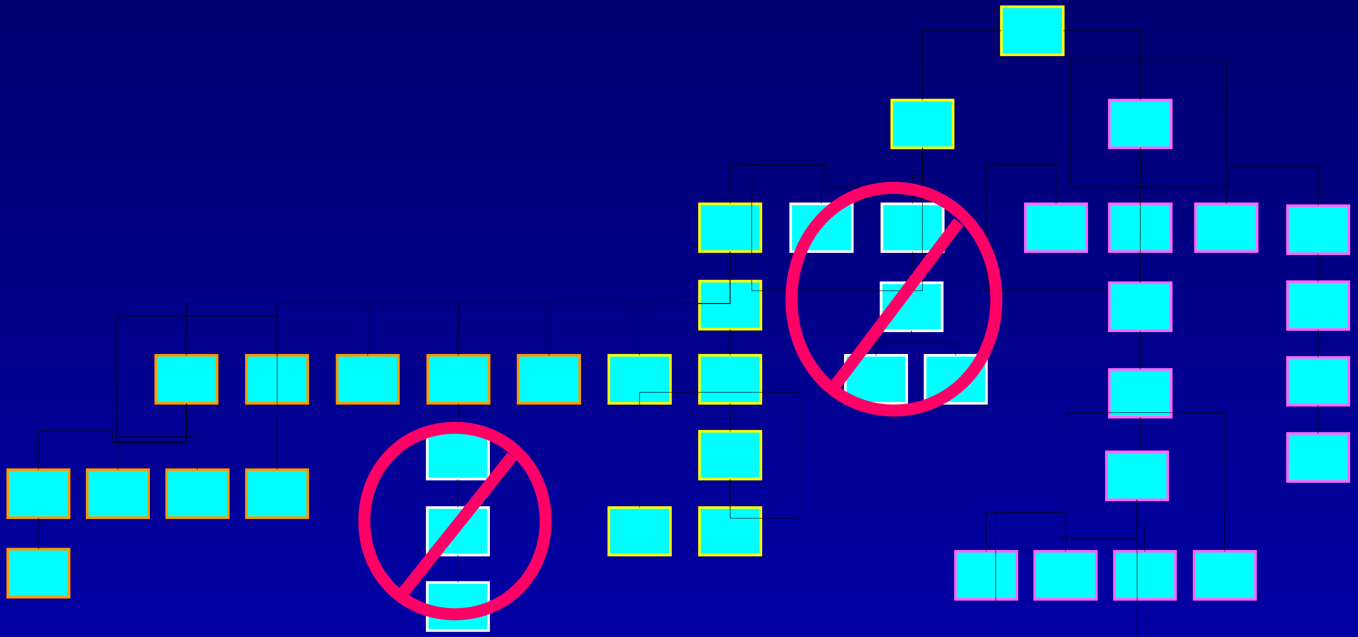
CYCORP

Discarding the Search Structure

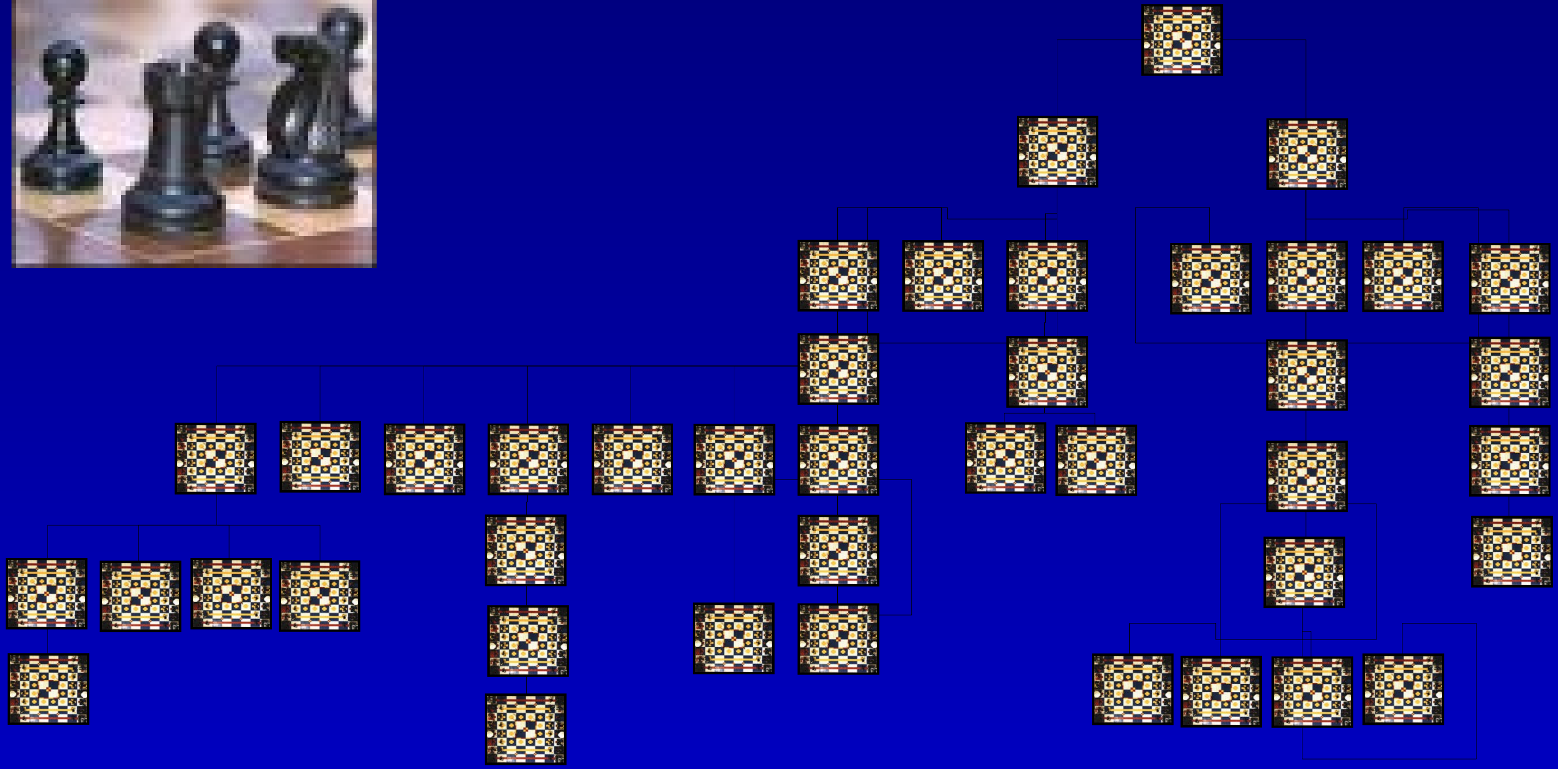
There is no need to save the proof of the states that border on Tennessee

Virginia
Missouri
Kentucky
North Carolina
Tennessee
Arkansas
Mississippi
Alabama
Georgia

The Search Tree is a Metaphor

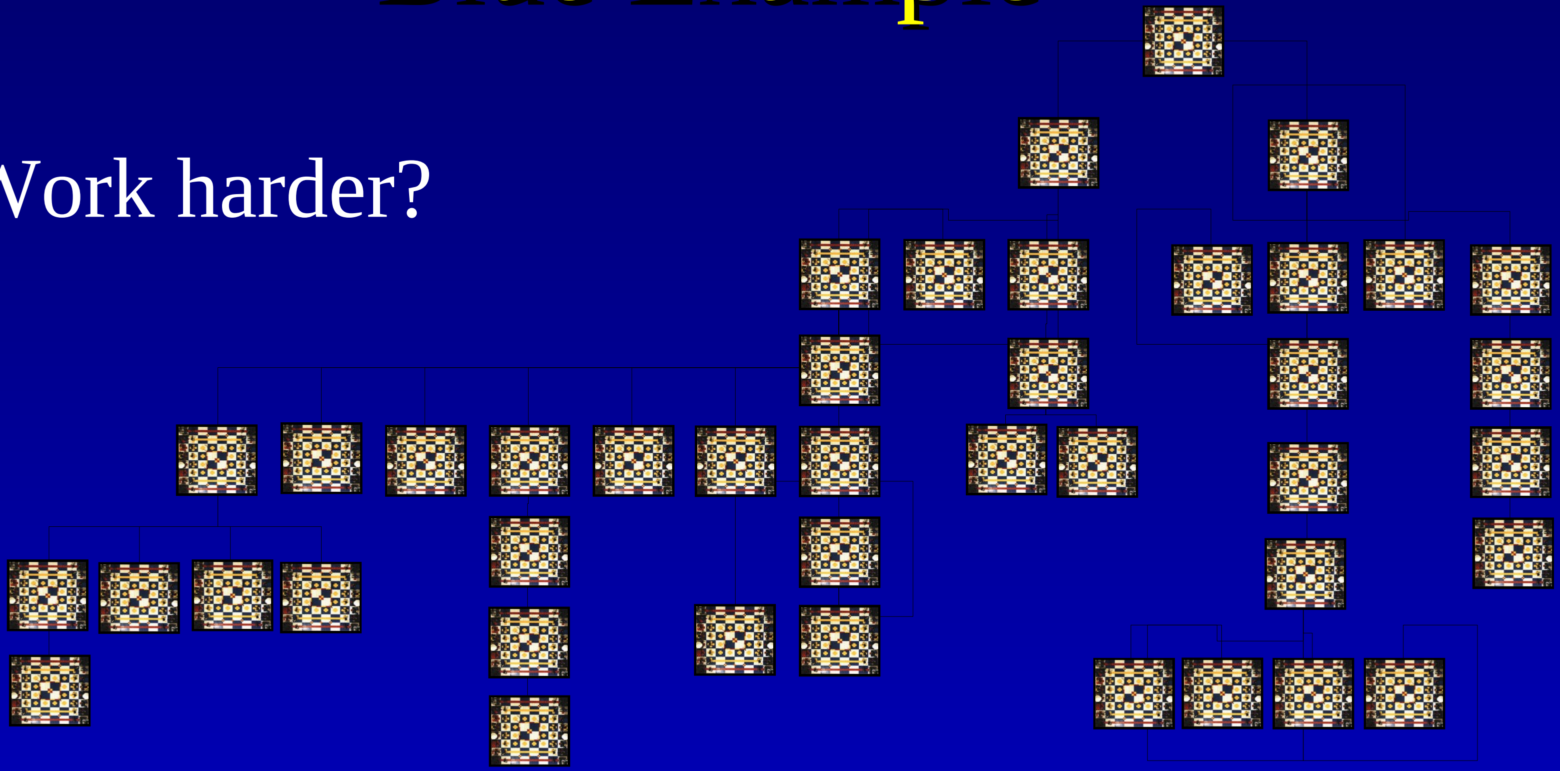


Work Harder or Smarter? Deep Blue Example



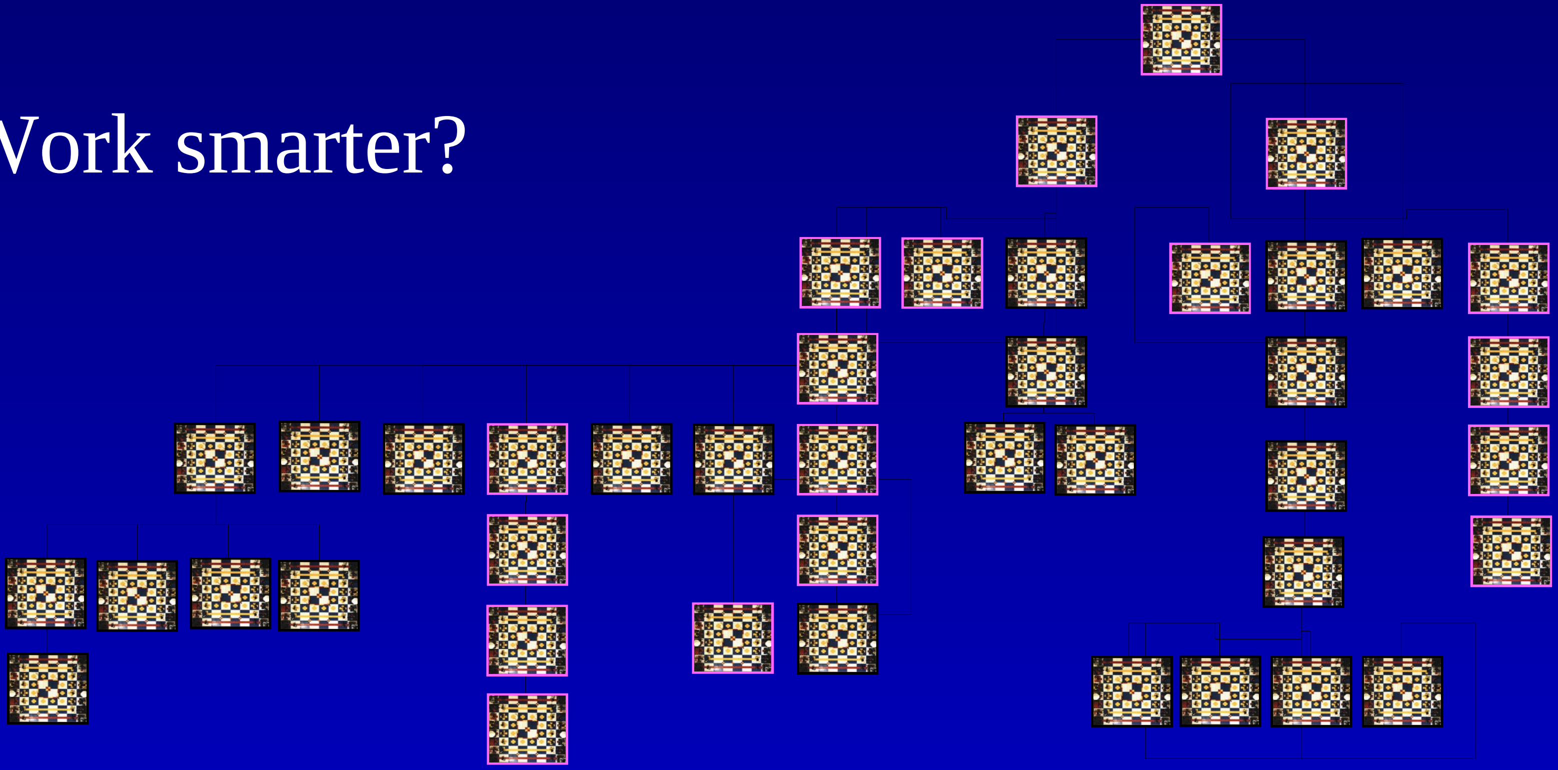
Work Harder or Smarter? Deep Blue Example

Work harder?



Work Harder or Smarter? Deep Blue Example

Work smarter?



Summary

- Life in the Big City
- Inference is Resource-bounded
- Resource-bounded Incompleteness
- Inference in Continuable
  - Proof Search is Stored
  - Meta data could be stored
- Deep Blue: Working Harder

CYCORP

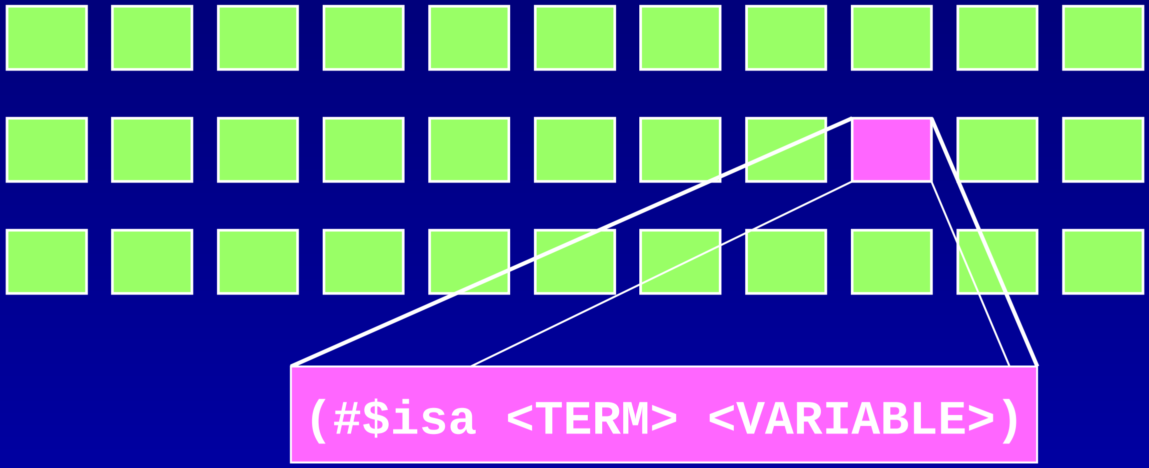
Inference in Cyc

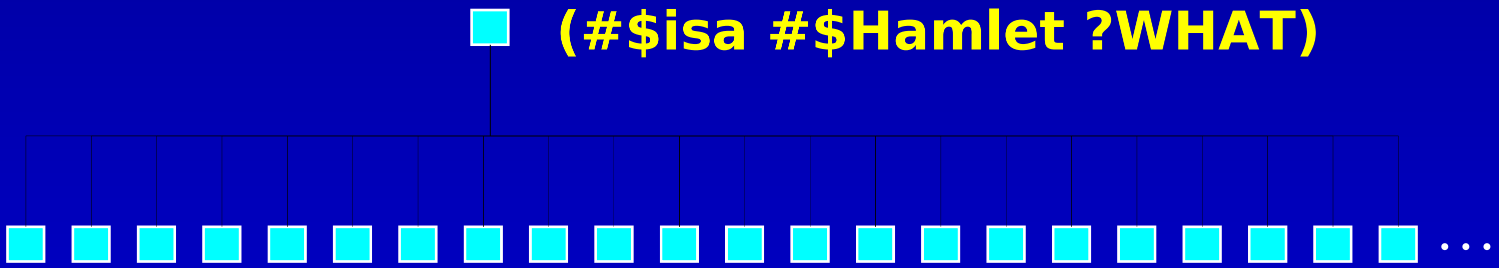
- Logical Aspects of Inference
- Incompleteness in Searching
- Incompleteness from Resource Bounds and Continuable Searches
- ***Efficiency through Heuristics***
- Inference Features in Cyc

CYCORP

Inference is Modular

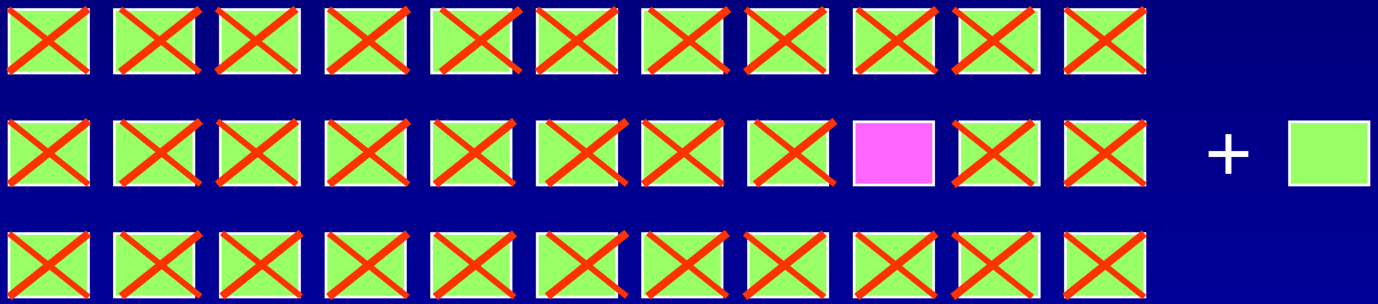
500+ special purpose inference modules





Inference is Modular (cont.)

Very flexible system can receive additions



Internal expert system does meta-reasoning:

“what are the inference options and modules that I have available to solve this problem?”

- quickly identify the irrelevant mechanisms ✗
- generate relevant child nodes

CYCORP

Inference is Bimodal

Mode 1:

Removal

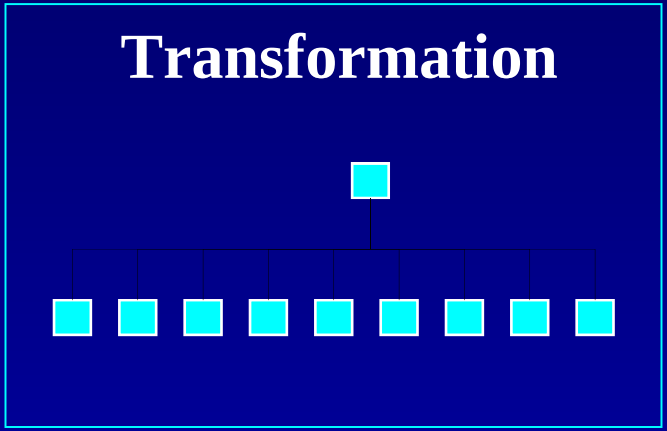
- simplifies problem
- decreases complexity
- use of facts

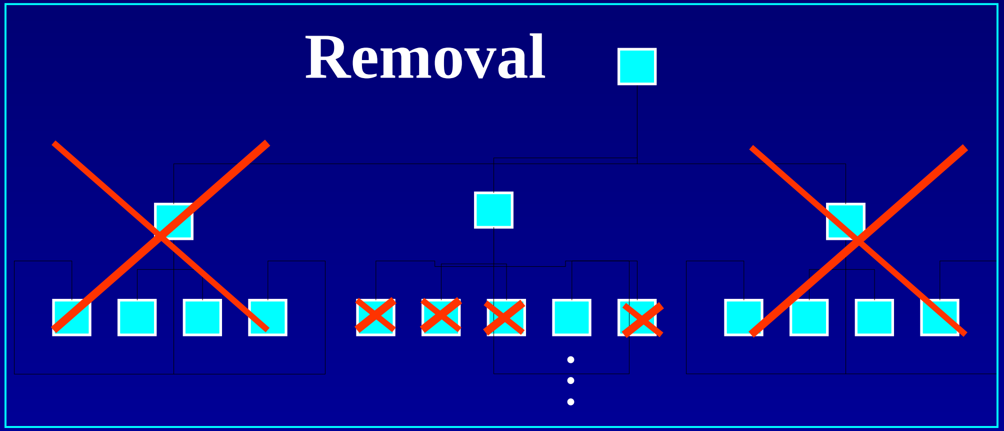
Mode 2:

Transformation

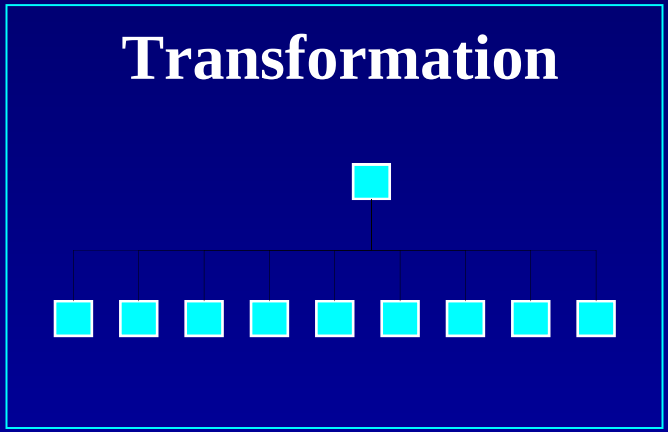
- transforms problem
- increases complexity
- use of rules

Inference is Bimodal (cont.)

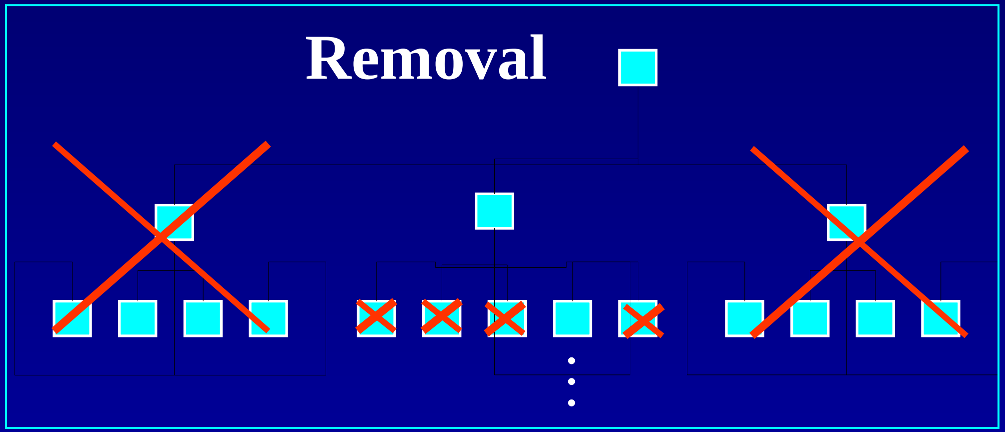




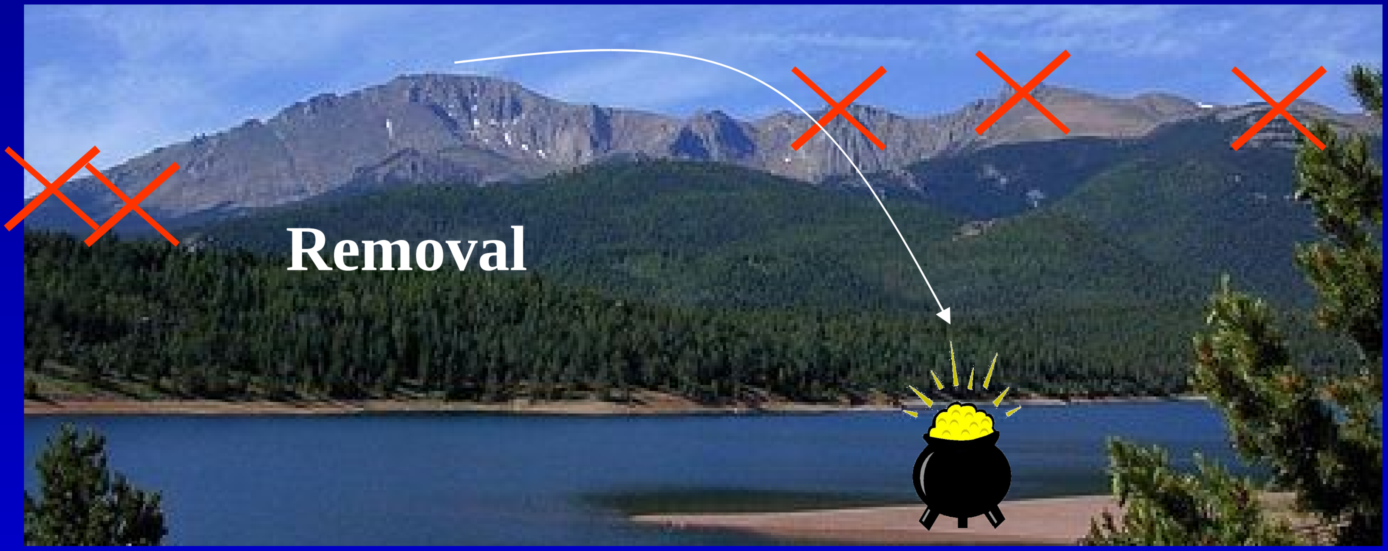
Inference is Bimodal (cont.)



Transformation



Removal



Removal

Inference is Heuristic

- Heuristics affect efficiency, not correctness

- Transformation heuristics
  - Order possible proofs based on rules used
  - Exhaustive queries require no ordering
- Removal heuristics
  - Generate answers as efficiently as possible
  - Prune dead-ends as early as possible

CYCORP

Summary

- Inference is Modular
  - Internal expert system does meta-reasoning
  - HL modules
- Inference is Bimodal
  - Removal (facts)
  - Transformation (rules)
- Inference is Heuristic
  - affect efficiency, not correctness

Inference in Cyc

- Logical Aspects of Inference
- Incompleteness in Searching
- Incompleteness from Resource Bounds and Continuable Searches
- Efficiency through Heuristics
- ***Inference Features in Cyc***

Inference Uses Mts for Consistency

<u>MainstreamAmericanCultureMt</u>

- (genls Vampire MythologicalThing)
- (isa LochNessMonster MythologicalThing)

In the MainstreamAmericanCultureMt,

- Vampire is a kind of mythological thing.
- The Loch Ness Monster is a mythological thing.

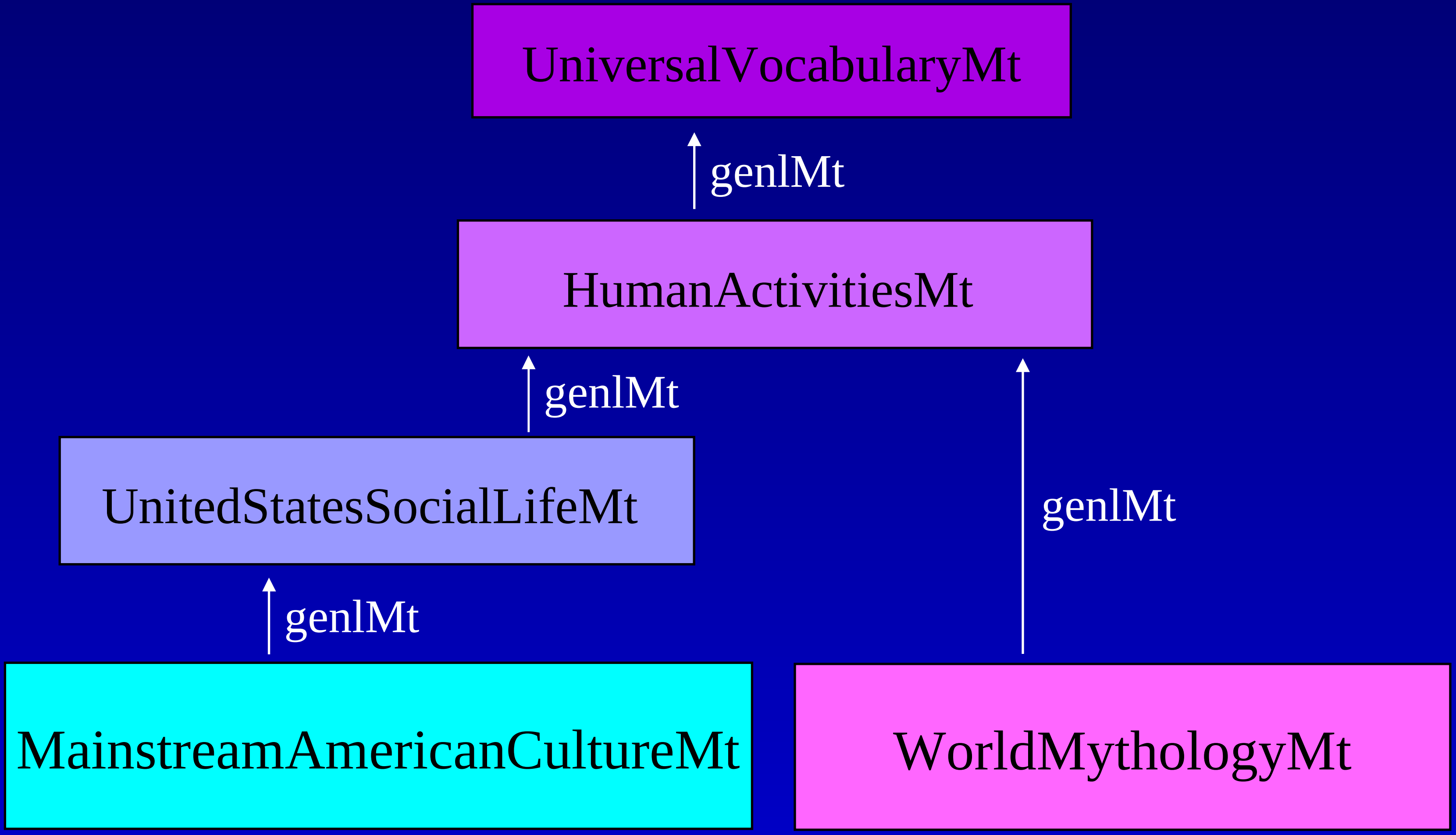
<u>WorldMythologyMt</u>

- (genls Vampire IntelligentAgent)
- (isa LochNessMonster Reptile)

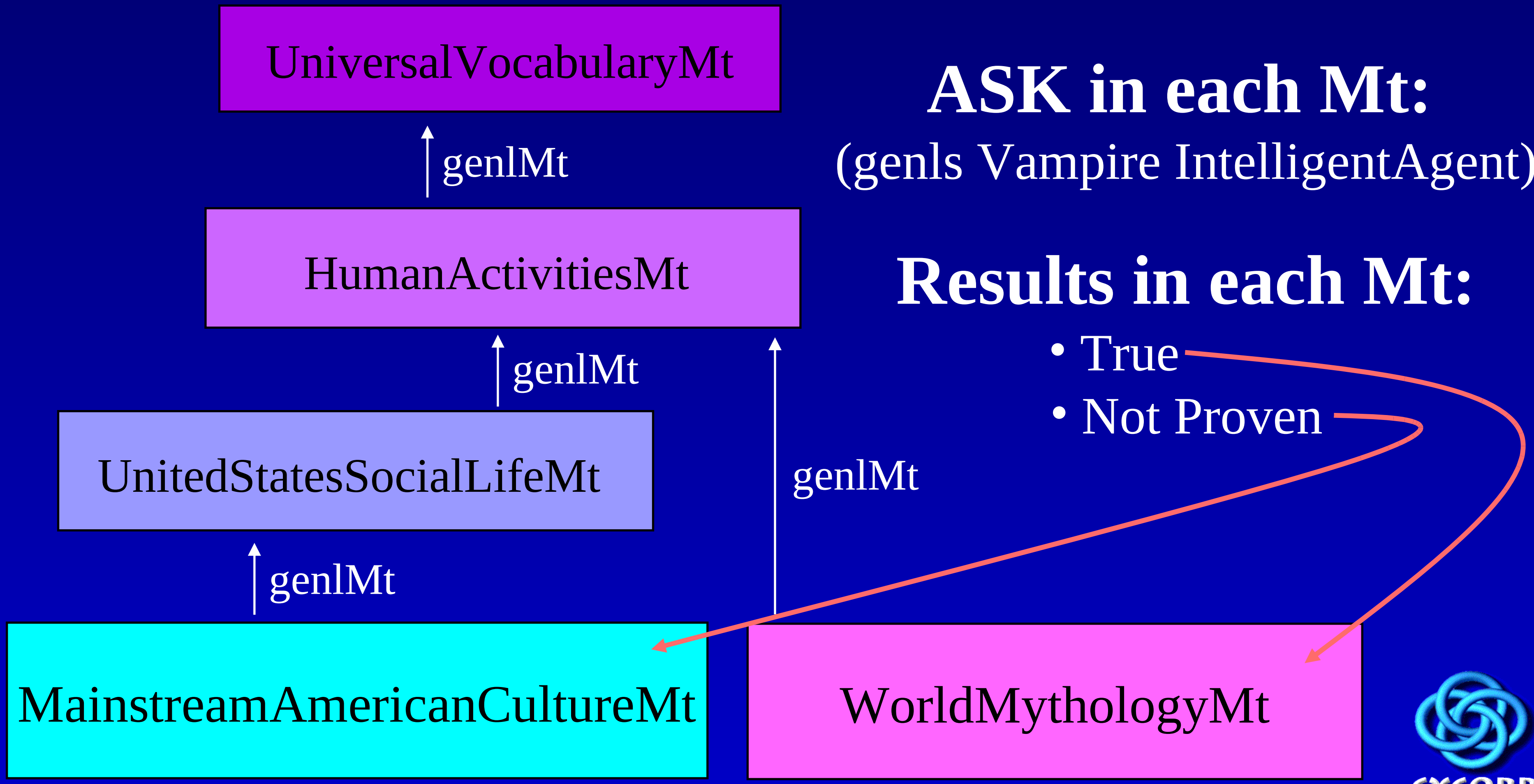
In the WorldMythologyMt,

- Vampire is a kind of intelligent agent.
- The Loch Ness Monster is a reptile.

Mts Inherit from More General Mts Using #$genlMt



Inference is performed *Within* Mts



UniversalVocabularyMt

genlMt

HumanActivitiesMt

genlMt

UnitedStatesSocialLifeMt

genlMt

genlMt

MainstreamAmericanCultureMt

WorldMythologyMt

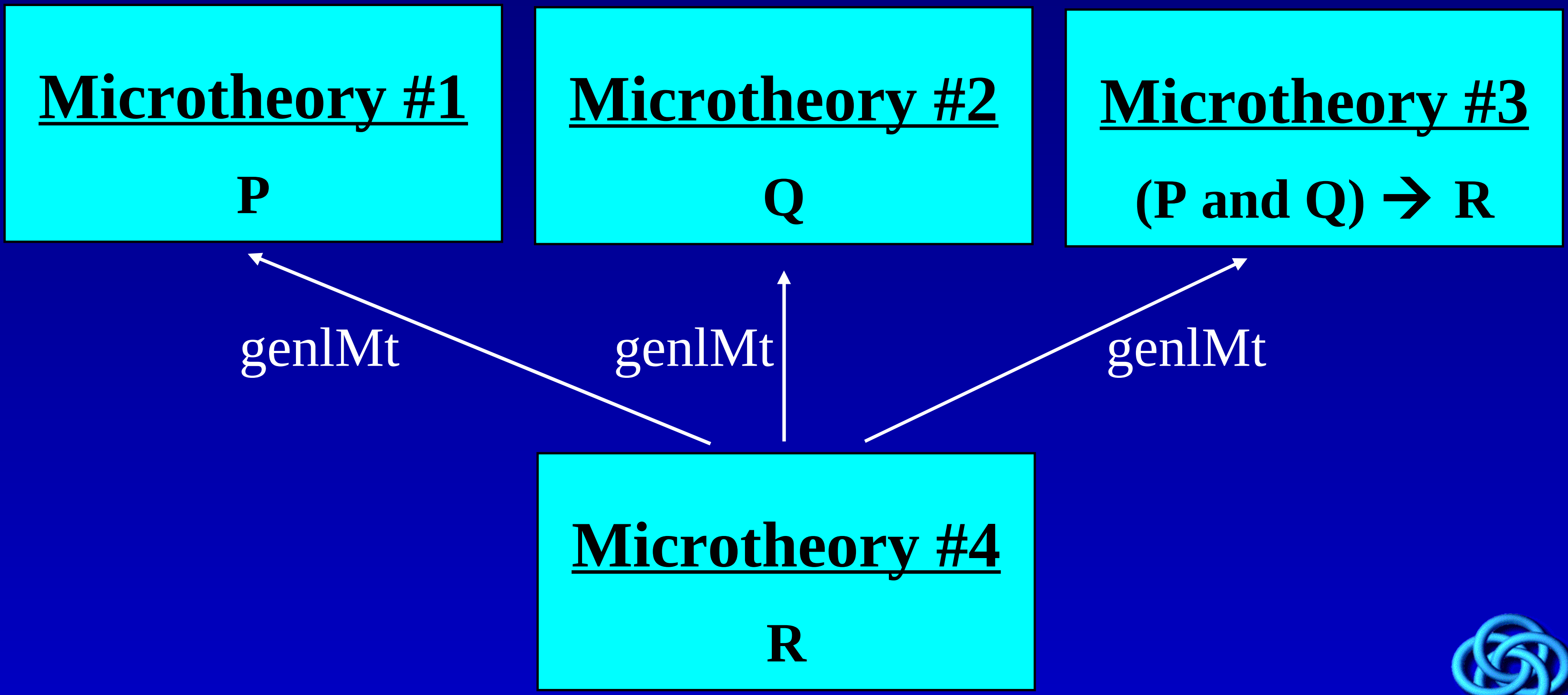
ASK in each Mt:

(genls Vampire IntelligentAgent)

Results in each Mt:

- True
- Not Proven

Inference Uses Microtheories and Inheritance



Microtheory #1

P

Microtheory #2

Q

Microtheory #3

(P and Q) → R

genlMt

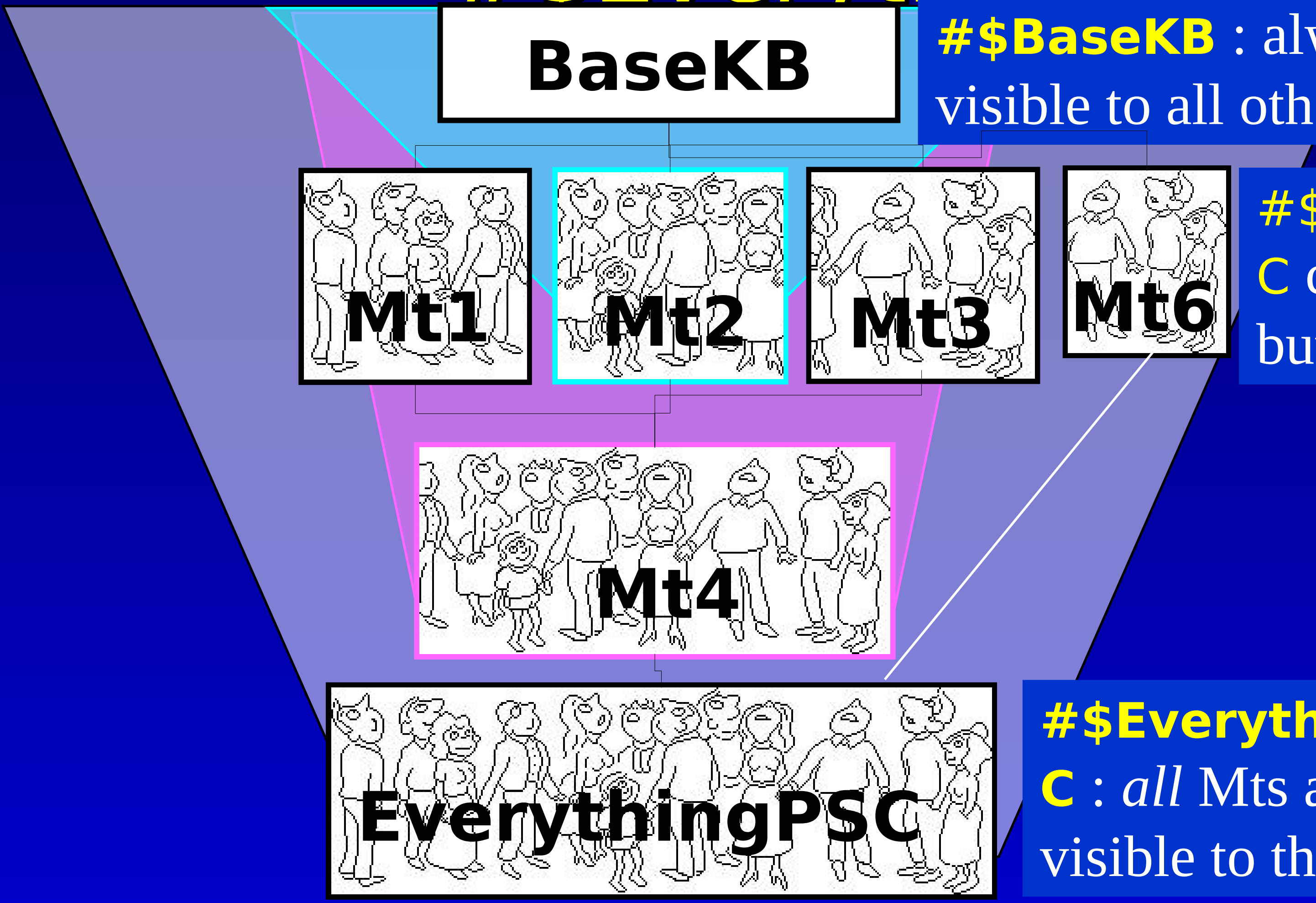
genlMt

genlMt

Microtheory #4

R

Two Important Microtheories: #$BaseKB and #$EverythingPSC

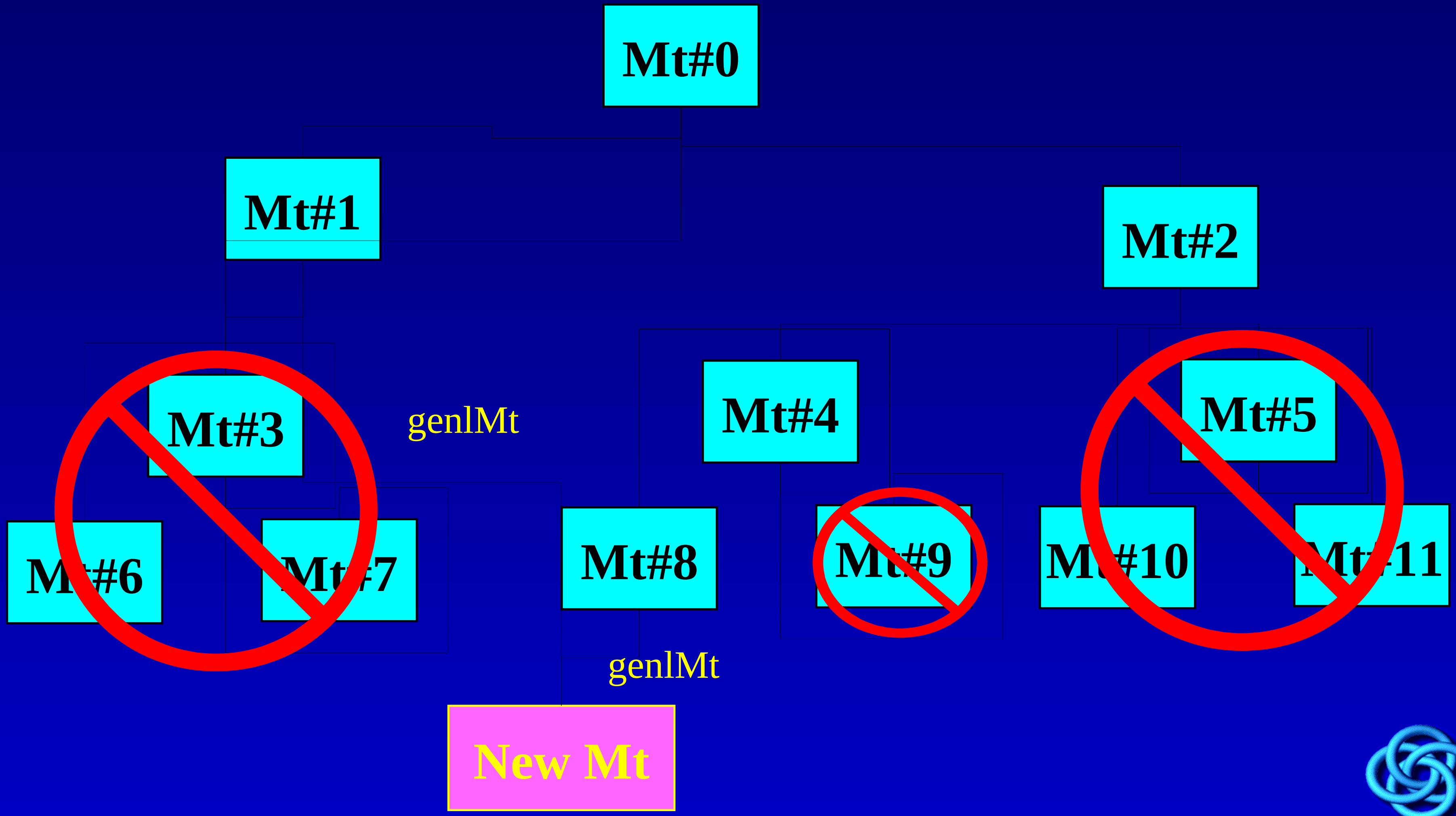


#$BaseKB : always visible to all other Mts

#$EverythingPSC can “see” Mt6, but Mt4 cannot.

#$EverythingPSC : *all* Mts are visible to this Mt

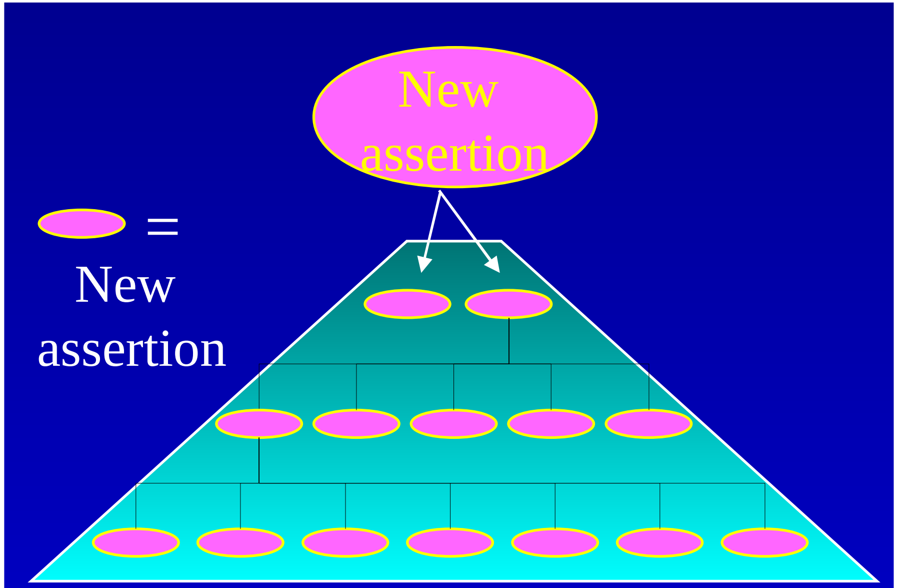
Placing a New Microtheory



Inference can be Forward or Backward

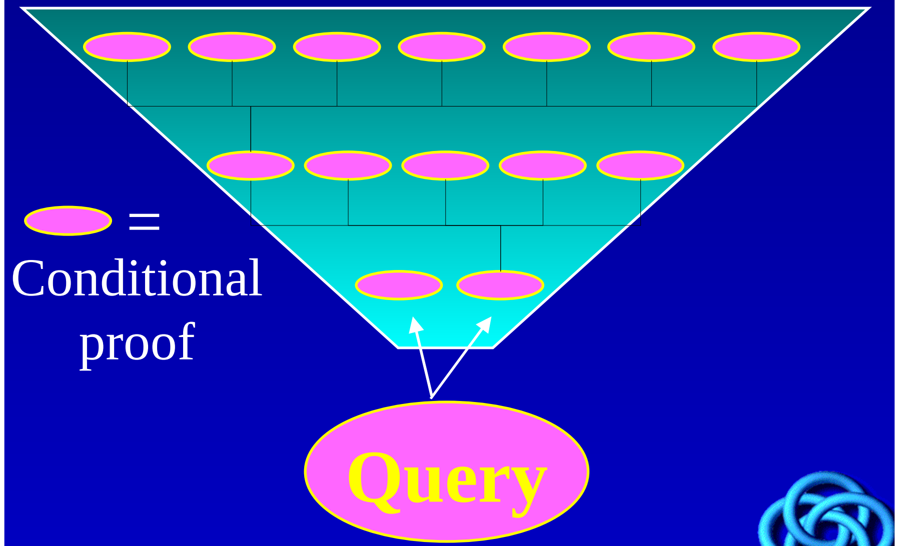
Forward Inference:

- Occurs at UPDATE time
- Causes new assertions to be added throughout the KB



Backward Inference:

- Occurs at QUERY time
- Creates conditional proofs to be proven by existing facts



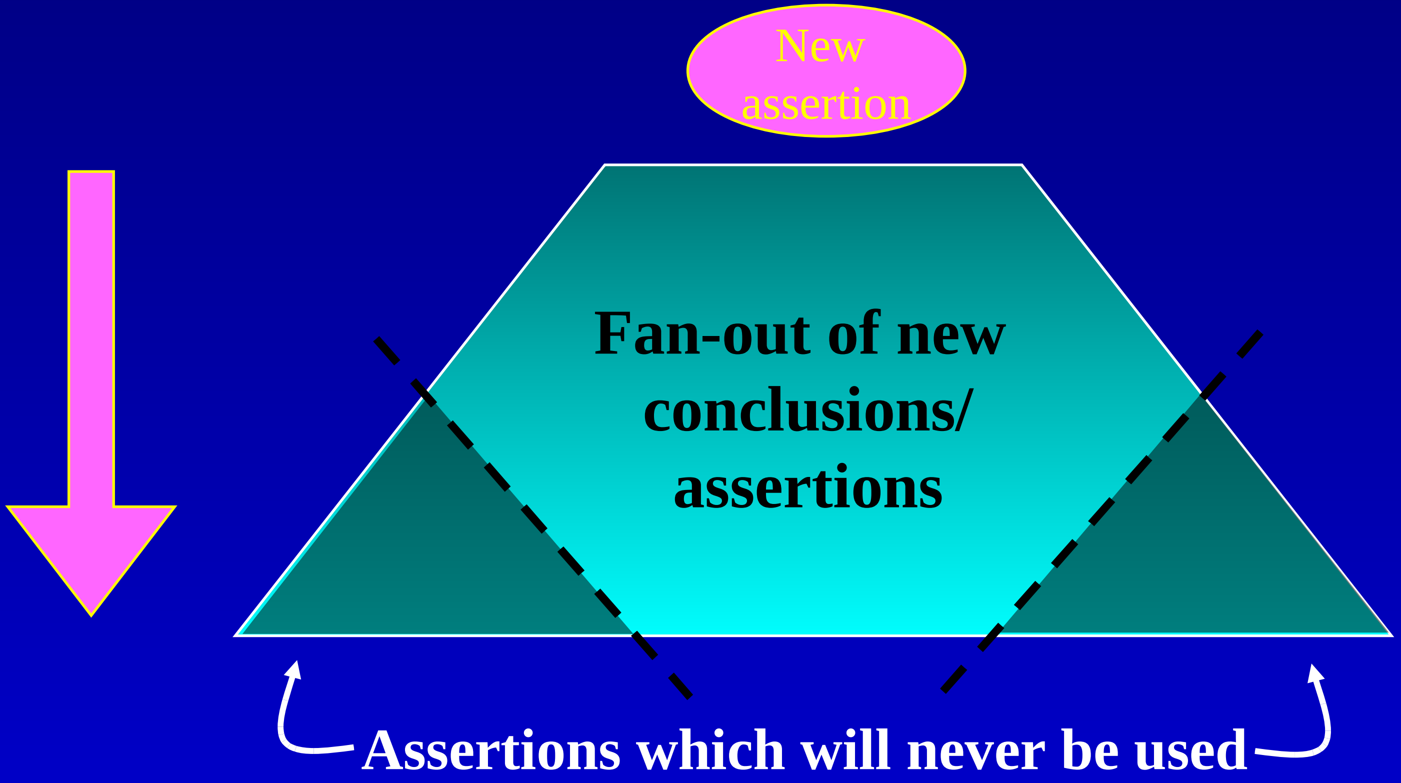
Forward Inference: Strengths and Weaknesses

Forward Inference: At assert time, eagerly attempt to provide a deductive chain between what you’re asking and what is already known.

+	-
• larger target for your backward inference to eventually hit	• a lot of work at update time • wasted effort in making new conclusions

Limitation of Forward Inference

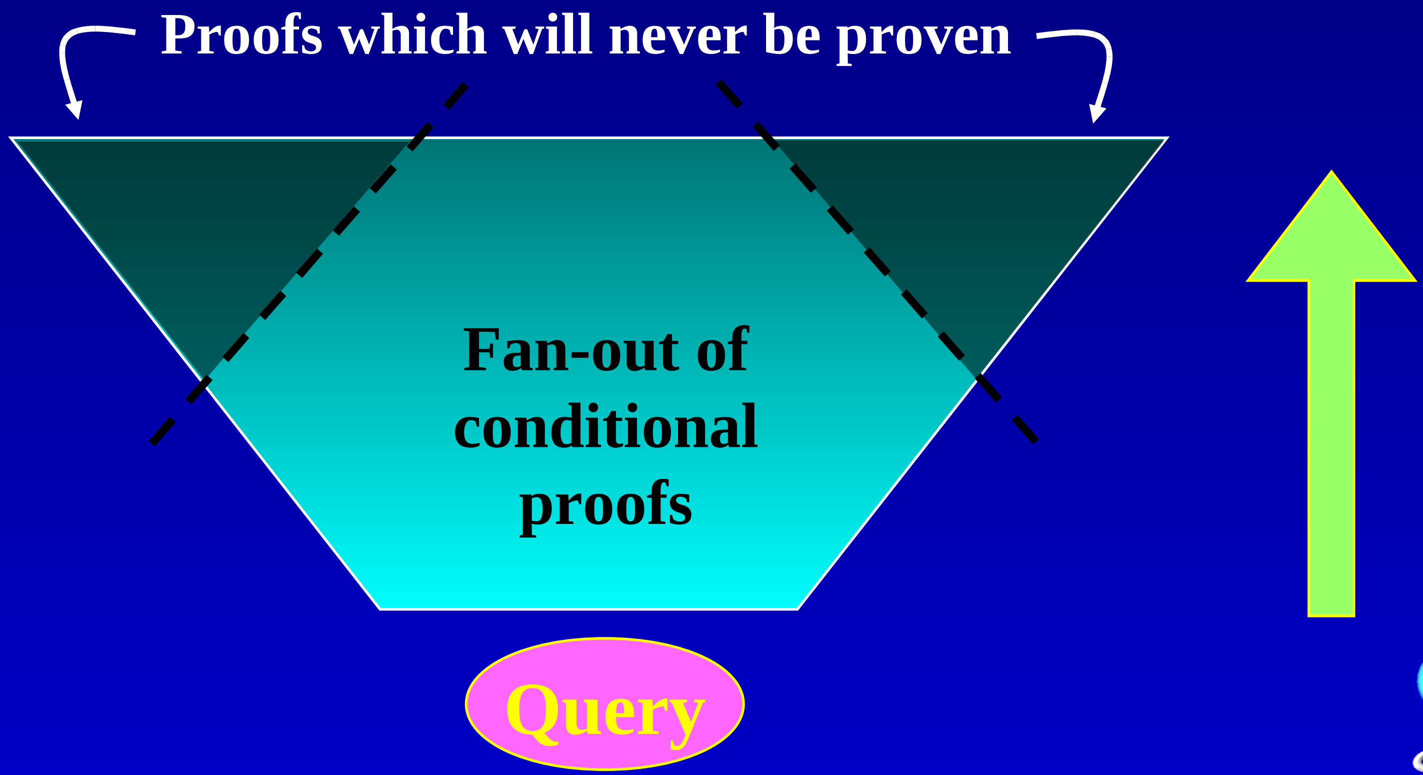
There is a certain size of knowledge base beyond which the space of conclusions you get in a forward fashion is so large that it just becomes unwieldy.


New assertion
Fan-out of new conclusions/ assertions
Assertions which will never be used

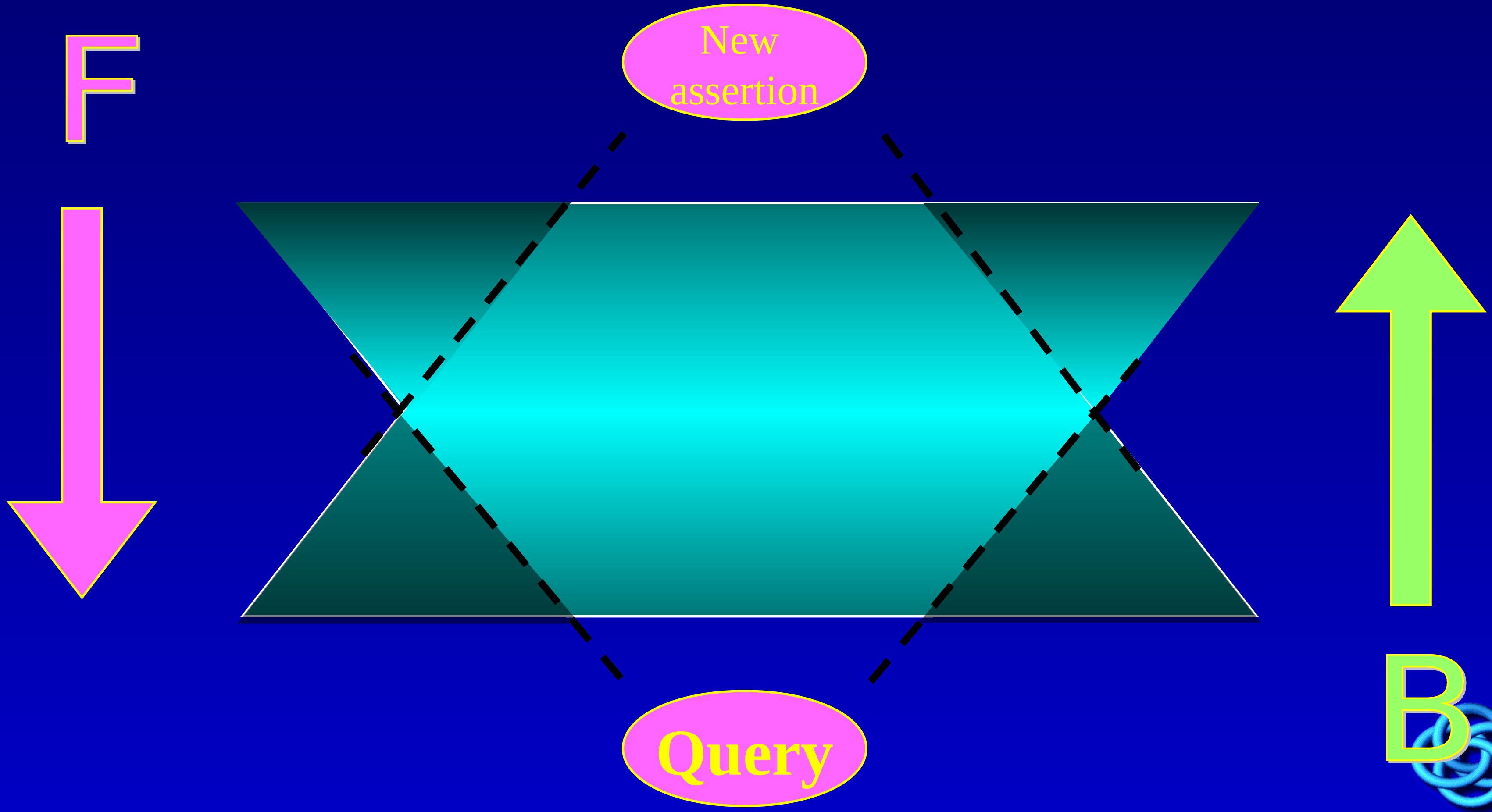
CYCORP

Limitation to Backward Inference

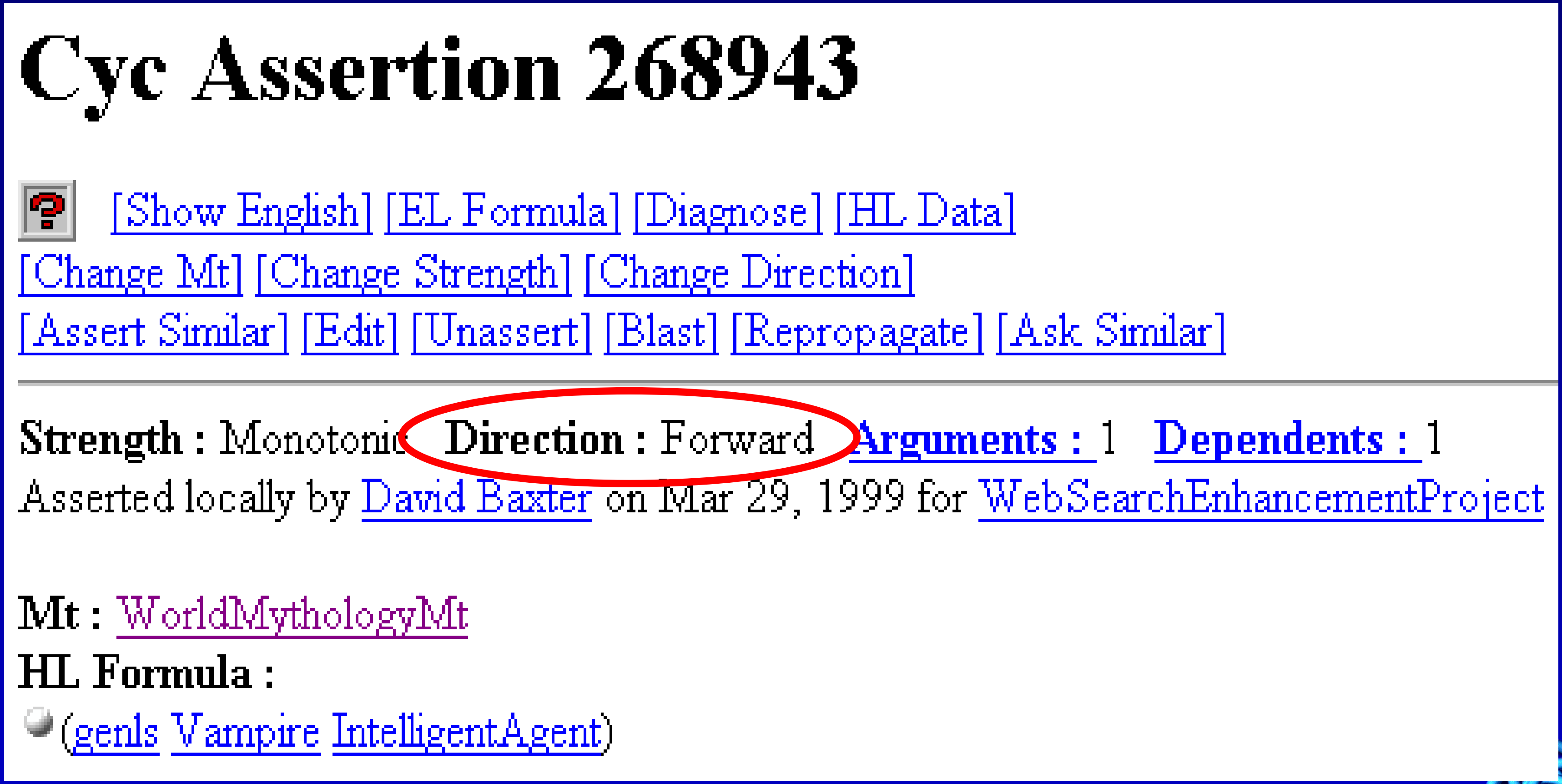
You can have enormous fan-out in the space of proofs which you are trying to prove which have no hope of ever targeting anything that is stated in your system.


Proofs which will never be proven
Fan-out of conditional proofs
Query

Cyc Supports Both Forward and Backward Inference



A Subset of the KB is Marked “Forward”



Cyc Assertion 268943

[Show English] [EL Formula] [Diagnose] [HL Data]
[Change Mt] [Change Strength] [Change Direction]
[Assert Similar] [Edit] [Unassert] [Blast] [Repropagate] [Ask Similar]

Strength : Monotonic **Direction :** Forward **Arguments :** 1 **Dependents :** 1

Asserted locally by David Baxter on Mar 29, 1999 for WebSearchEnhancementProject

Mt : WorldMythologyMt

HL Formula :

(genls Vampire IntelligentAgent)

Copyright © 2002 Cycorp

Summary

- Inference Uses Mts for Consistency
- Mts Inherit from More General Mts Using #$genlMt
- Inference is performed Within Mts
- Two Important Microtheories: #$BaseKB and #$EverythingPSC
- Inference can be Forward or Backward
- A Subset of the KB is Marked “Forward”