

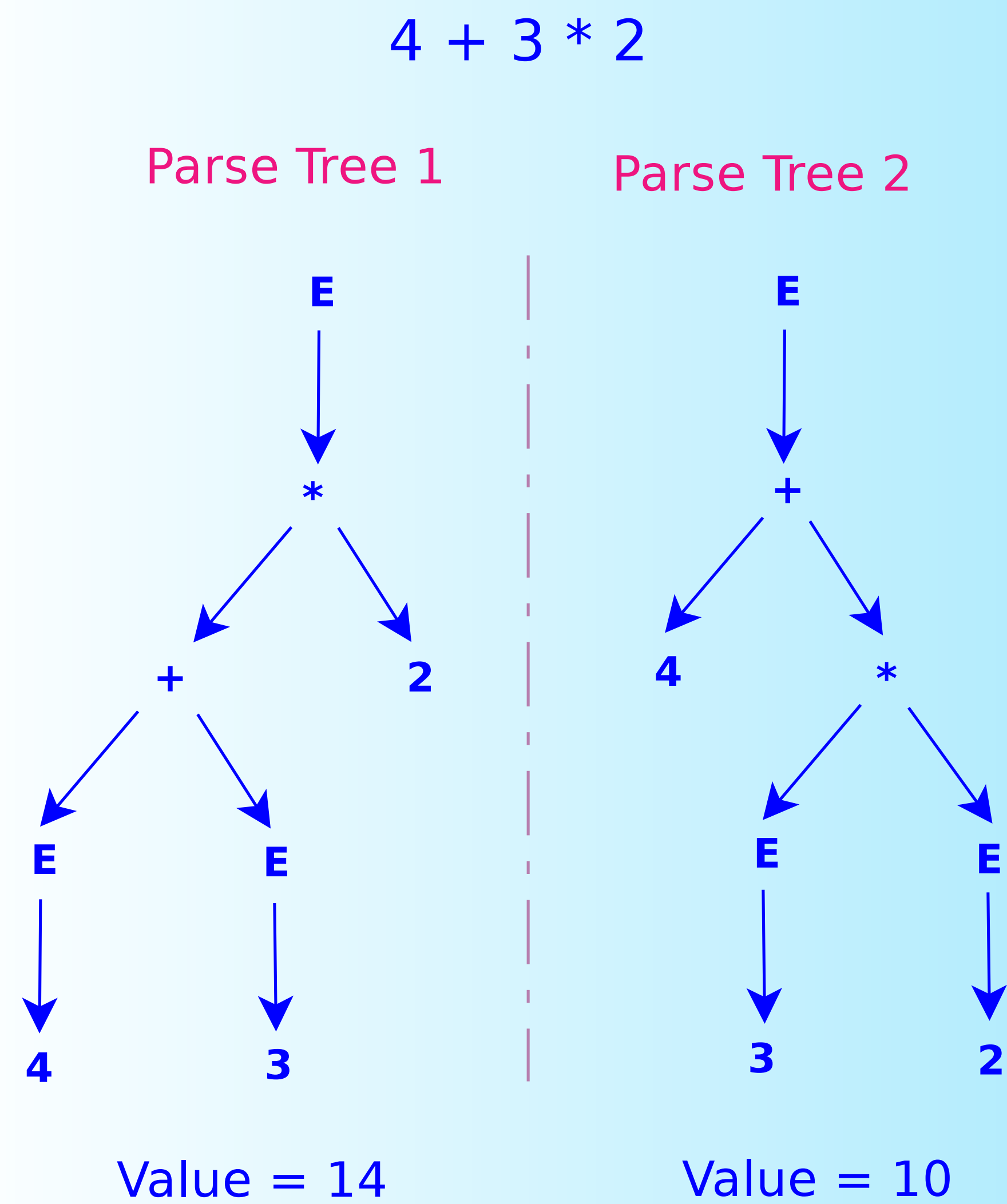
# SEARCH-BASED AMBIGUITY DETECTION IN CONTEXT-FREE GRAMMARS

NAVENEETHA VASUDEVAN, LAURENCE TRATT  
King's College London, Strand, London, WC2R 2LS, United Kingdom

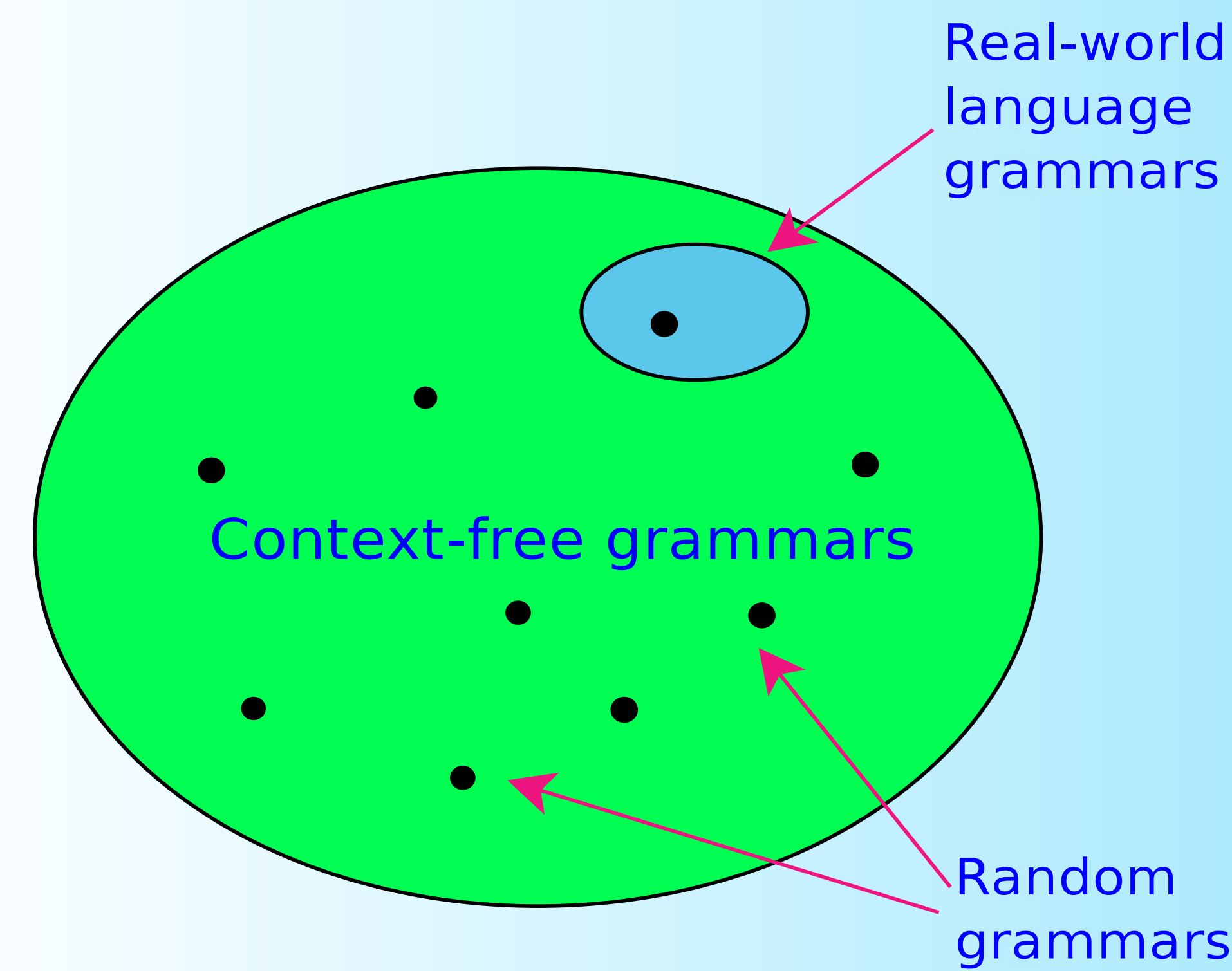
## AMBIGUITY PROBLEM

1. Ambiguity is an undecidable problem [1]
2. Multiple parse trees are undesirable

$$E \rightarrow E' + E \mid E' * E$$

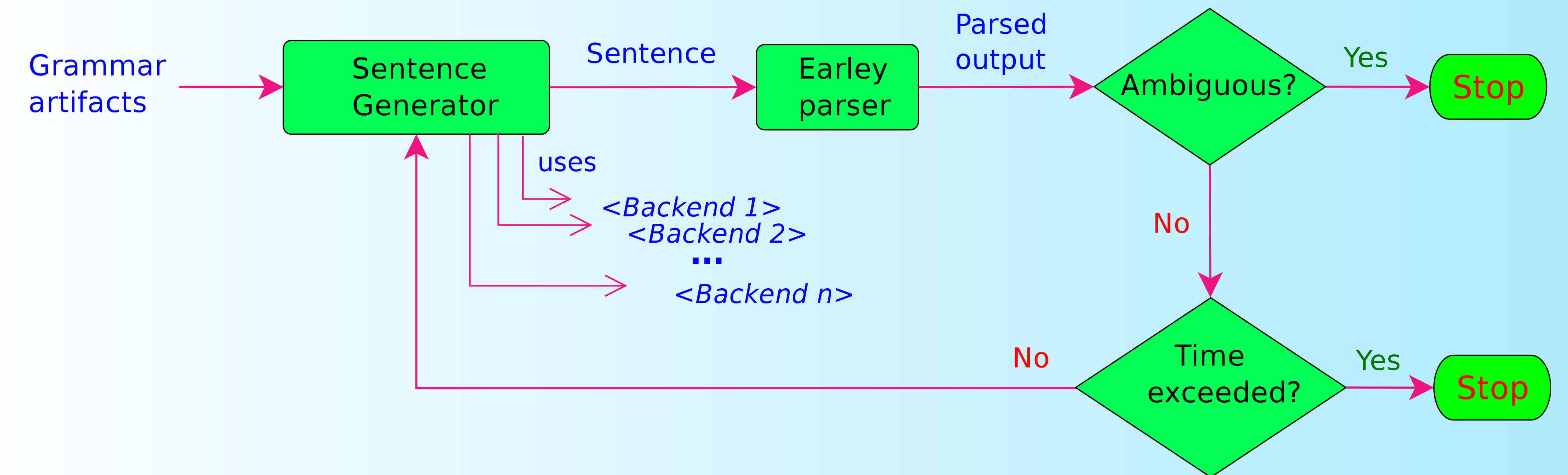


## CFGs



- Real-world grammars – A tiny subset of entire class of CFGs
- Random grammar generator – We aim that the grammars generated span a much wider subset of CFGs

## SINBAD



- SinBAD is a search-based tool to ambiguity detection
- The *Sentence Generator* component generates random sentences using a *backend* instance. A backend, in essence, is an algorithm that governs how sentences are generated
- An Earley-based parser is used to check for ambiguity
- The search stops when an ambiguity is found or when a time limit is exceeded

## EXPERIMENT

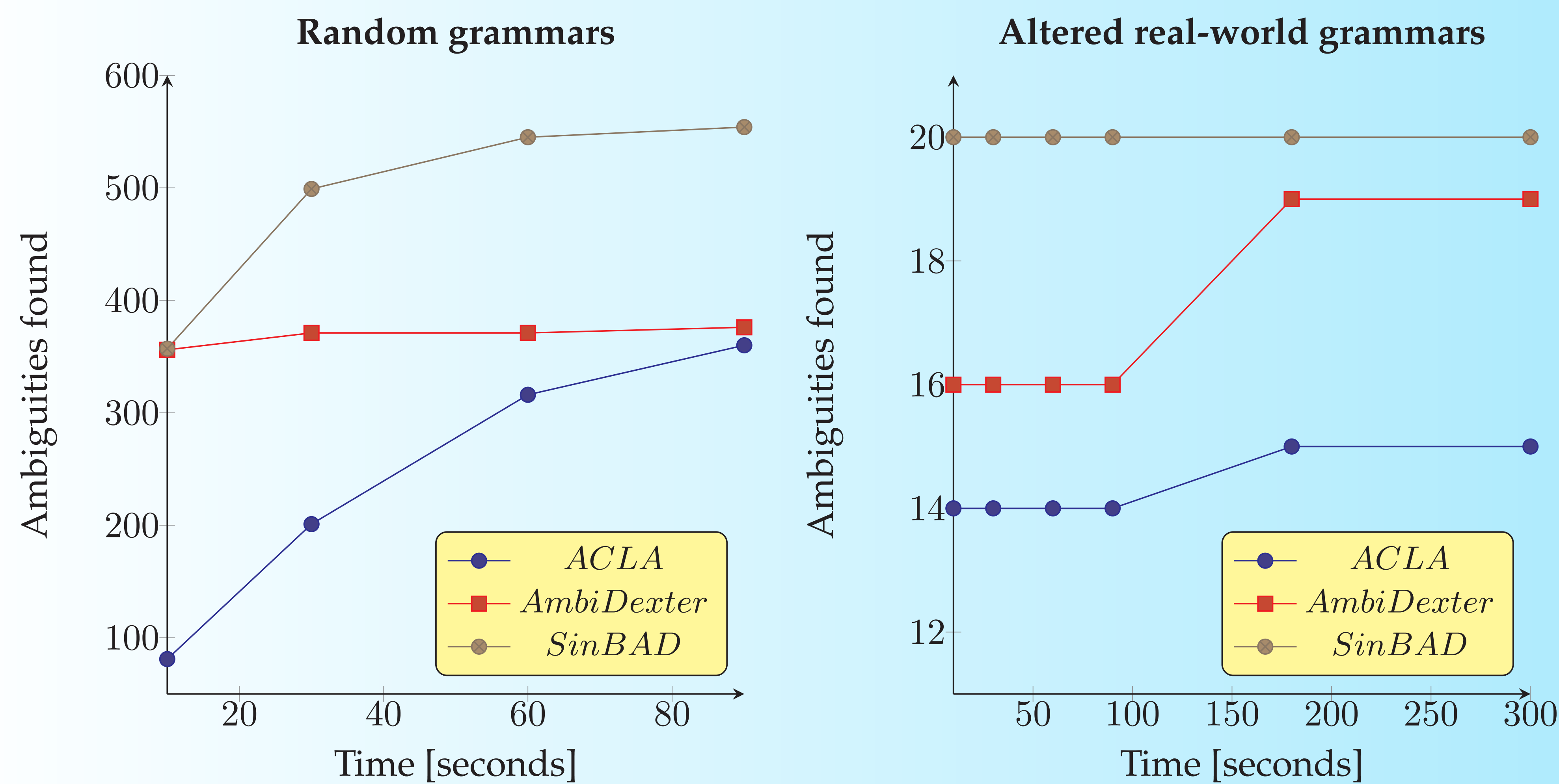
To understand how well our search-based approach uncovers ambiguity. We evaluate SinBAD against two tools:

1. ACLA - uses an approximation technique
2. AmbiDexter - uses a hybrid approach (grammar filtering + search)

Grammar sets include:

1. Random grammars (tested for 10, 30, 60, & 90s)
2. Altered real-world grammars [2] (tested for 180 & 300s)

## RESULTS



## FUTURE DIRECTIONS

- To add more tools to the study and perform a larger experiment with more real-world-esque grammars
- To develop a CFG equivalent of the work on random generation of automata [3]

## SOURCE CODE

The source code for SinBAD is available at <http://github.com/nvasudevan/sinbad>

## REFERENCES

- [1] David G. Cantor. On the Ambiguity Problem of Backus Systems. In *ACM '62*
- [2] H.J.S. Basten and J. J. Vinju. Faster Ambiguity Detection by Grammar Filtering. In *LDTA '10*
- [3] Héam, Pierre-Cyrille and Nicaud, Cyril and Schmitz, Sylvain. Random Generation of Deterministic Tree (Walking) Automata. In *CIAA '09*