

Program Synthesis for Character Level Language Modeling

Pavol Bielik, Veselin Raychev, Martin Vechev



Character Level Language Model

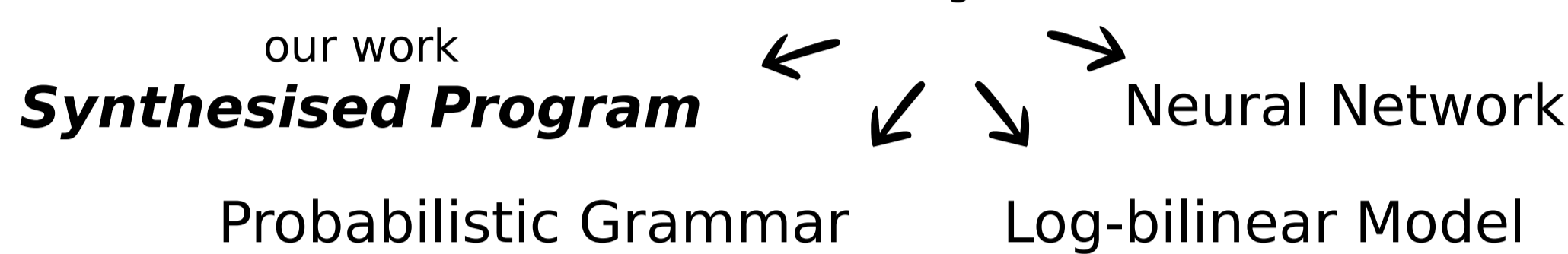
Statistical language model that estimates a probability distribution over sequences of characters from data

the brown fox jumps over the lazy dog
 x_1 x_n

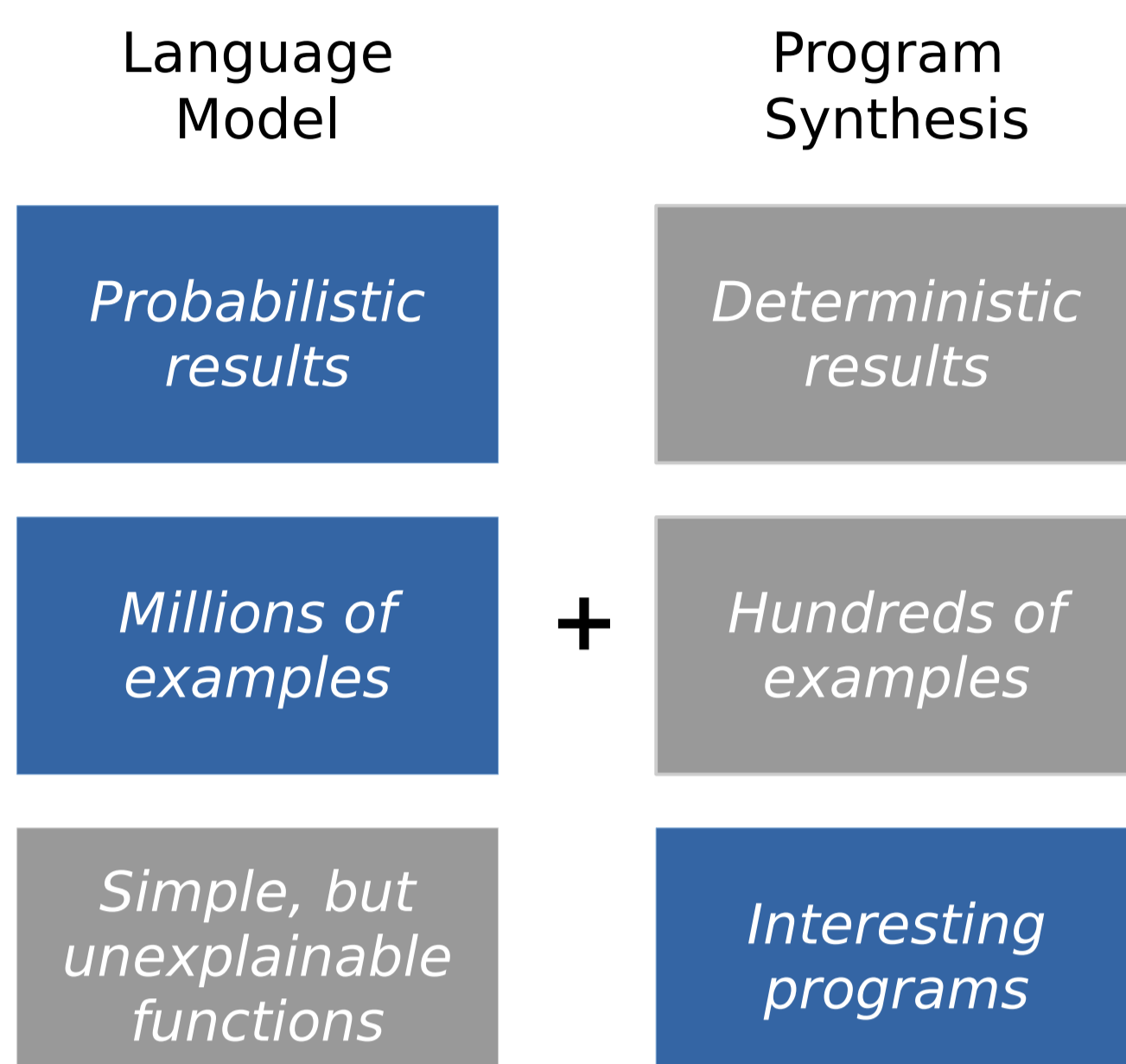
Generalized problem statement:

$$\arg \min_{\theta} \frac{1}{n} \sum_{t=1}^n \log p(x_t | f(x_{<t}, \theta))$$

instantiating function f

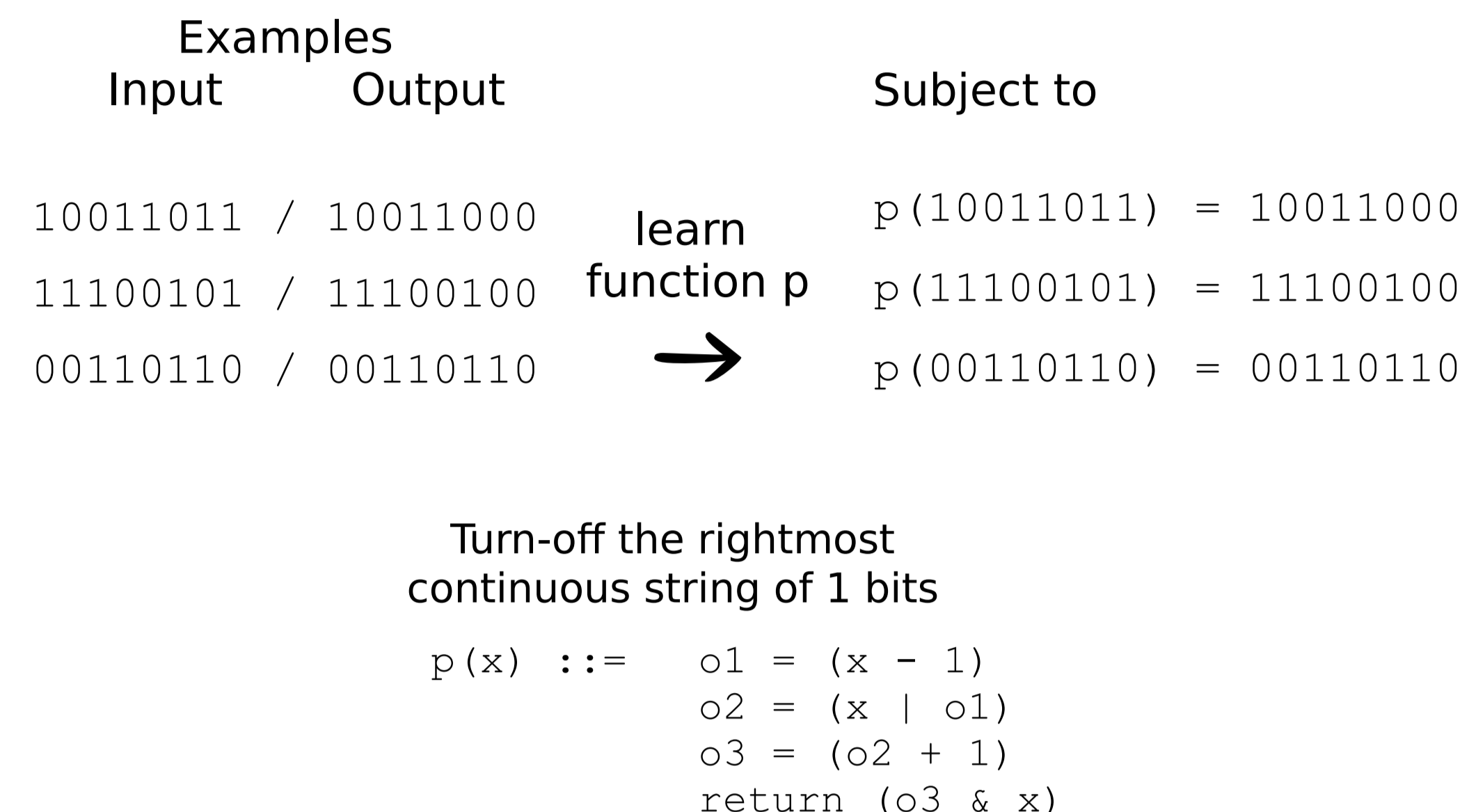


Our Work



Program Synthesis

Automatically constructs a program that satisfies a given specification (e.g., input/output examples)

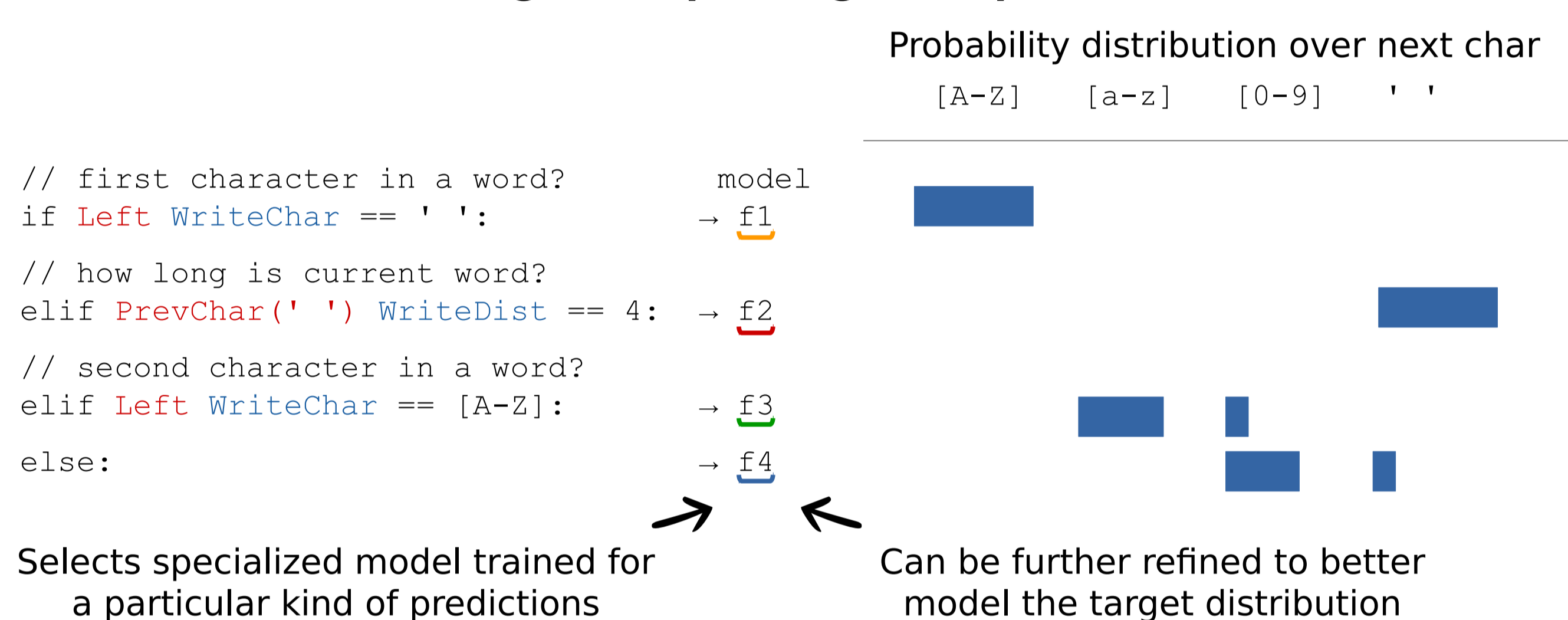


Using Programs to Explain Data

Input (sequence of atoms)

' Mg12 He2 Ai13 Fe26 Mg12 Ag47 ... '

Program explaining the input



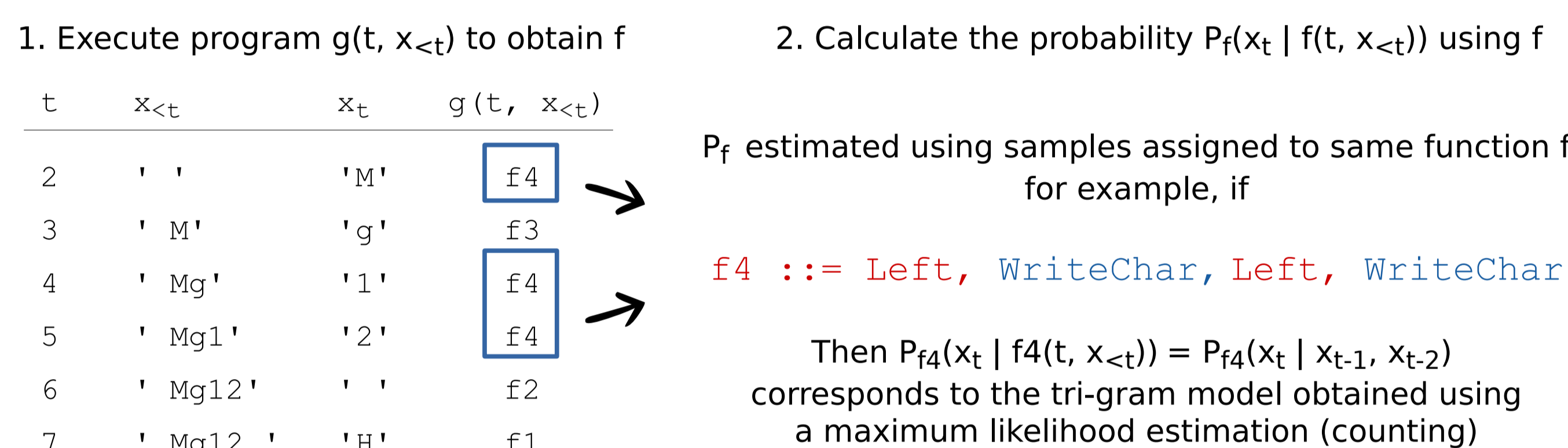
What is the model f ?

In general: Any model (e.g., Neural Network)

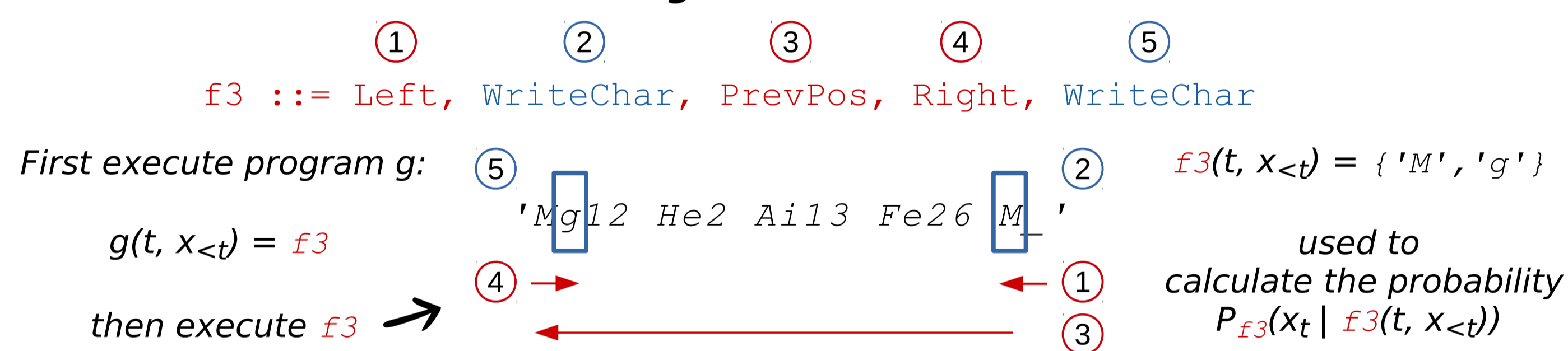
Our work: Another program (LMProgram)

Obtaining Probabilistic Model from a Program

Two step process to obtain a probabilistic model from a learned program g



Program execution



DSL for Character Level Language Modeling

Expresses non-trivial data dependencies using instructions that operate on a sequence of characters

Allows conditioning on previously seen input

Allows conditioning on program state

TChar ::= SwitchProgram | StateProgram | return LMProgram

LMProgram ::= SimpleProgram

| SimpleProgram backoff d; LMProgram

| (SimpleProgram, SimpleProgram)

Different model used if current model is not confident in a prediction

Predict a value taken from a previous position in the input

Basic instructions that operate on a sequence of characters

SimpleProgram ::= ϵ | Move; SimpleProgram | Write; SimpleProgram

Move ::= Left, Right, PrevChar($c \in \text{Vocabulary}$), PrevPos

Write ::= WriteChar, WriteHash, WriteDist

Learning a Program

Search technique

SimpleProgram

SwitchProgram

Enumerative search + MCMC

ID3+ algorithm

Raychev, V. et. al. Learning Programs from Noisy Data. POPL '16, ACM

This work

Raychev, V. et. al. Probabilistic Model for Code with Decision Trees. OOPSLA '16, ACM

Problem statement:

$$\arg \min_{g \in \text{TChar}} \frac{1}{n} \sum_{t=1}^n \log p(x_t | f(t, x_{<t})) + \lambda \cdot \Omega(g)$$

Regularization to avoid too complex programs

Evaluation on Source Code and Natural Language

Datasets

Markus Hutter <http://prize.hutter1.net/>

Hutter Prize Wikipedia (Natural Language + Metadata)

characters	vocabulary size
100 000 000	205

Karpathy, A. et. al. Visualizing and understanding recurrent networks. ICLR 2016 Workshop

Linux Kernel (Source Code + Comments)

characters	vocabulary size
6 206 996	101

Hutter Prize Wikipedia Dataset

Metric	n-gram	This work	Graves et. al. 2013	Sutskever et. al. 2011	We et. al. 2016	Chung et. al. 2017
		DSL model	Stacked LSTM	MRNN	MI-LSTM	HM-LSTM
BPC	1.94	1.67	1.62	1.60	1.44	1.34

Linux Kernel Dataset

Model	Bits per Character	Error Rate	Training Time	Queries per Second	Model Size
LSTM (2x512)	2.05	38.1%	~80 Hrs	300	53 MB
n-gram (7-gram)	2.23	35.9%	4 Sec	41 000	24 MB
TChar _{w/o} cache & backoff	1.92	33.3%	~ 8 Hrs	62 000	17 MB
TChar _{w/o} backoff	1.84	31.4%	~ 8 Hrs	28 000	19 MB
TChar _{w/o} cache	1.75	28.0%	~ 8.2 Hrs	24 000	43 MB
TChar	1.53	23.5%	~ 8.2 Hrs	3 000	45 MB

Learned specialized programs for Linux Kernel Dataset

