

StiCProb: A Novel Feature Mining Approach using Conditional Probability

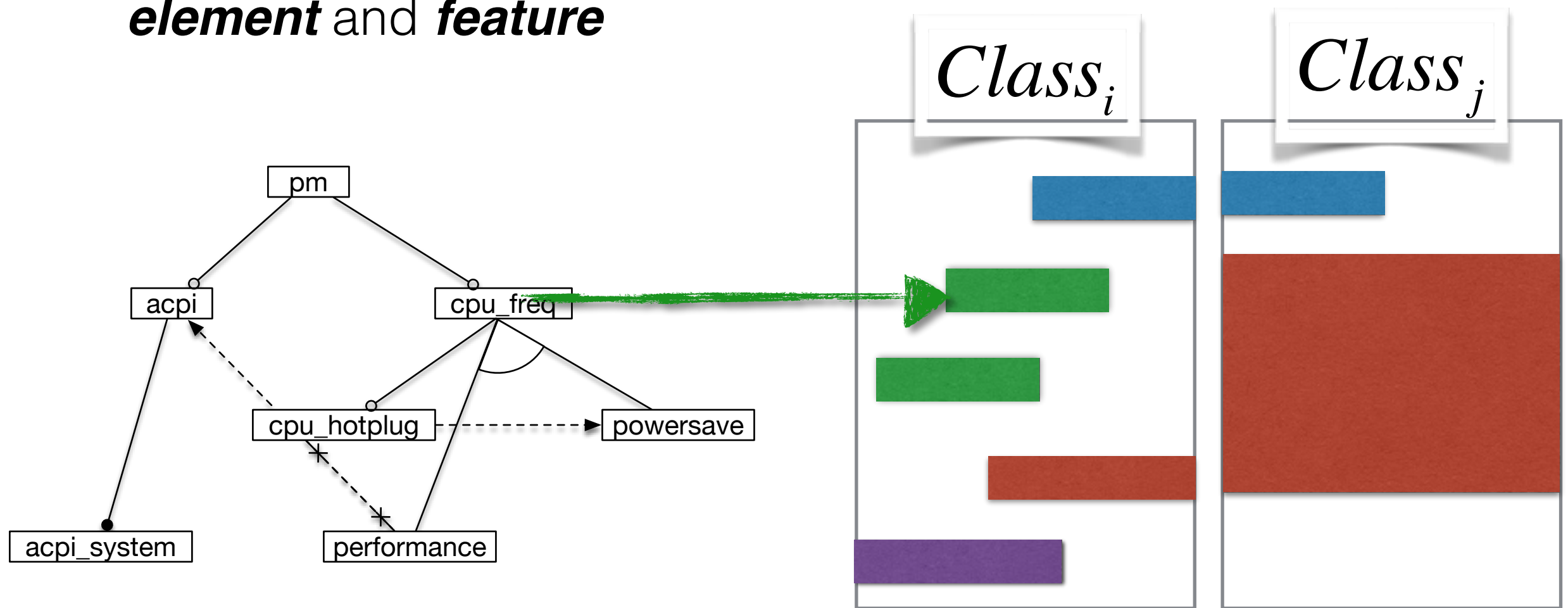
Yutian Tang, and Dr. Hareton Leung
The Hong Kong Polytechnic University
@SANER 2017

more: www.chrisyttang.org/loong



Motivation

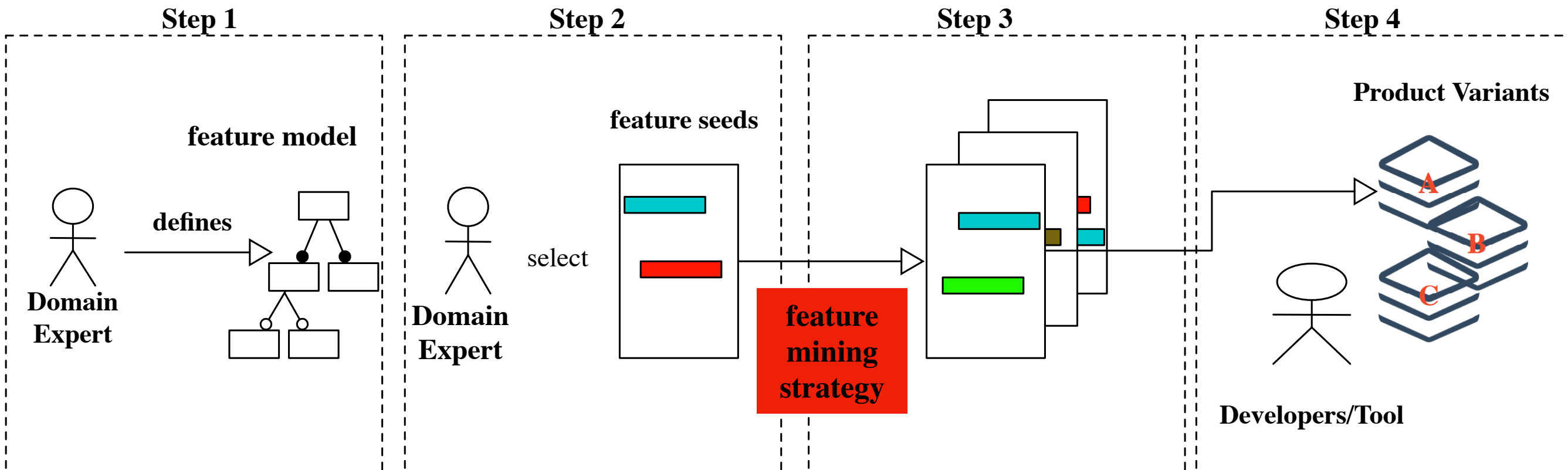
- Giving a correlation coefficient to model the closeness between ***element*** and ***feature***



1. How to locate the feature ?
2. How to measure ?

feature \longleftrightarrow *element*

How to locate?



1. Select seeds
2. Annotate features

Basic

Programming elements

E

Relationship

$$R \subseteq E \times E$$

Feature

F

(
Annotation

$$A \subseteq E \times F$$

Basic (cond')

mutual exclusion

$$M \subseteq F \times F$$

implications

$$\Rightarrow \subseteq F \times F$$

full annotation

$$A \subseteq E \times F$$

$$A^* = \left\{ (e, f) \mid (e, f) \in A, g \Rightarrow^* f \right\}$$

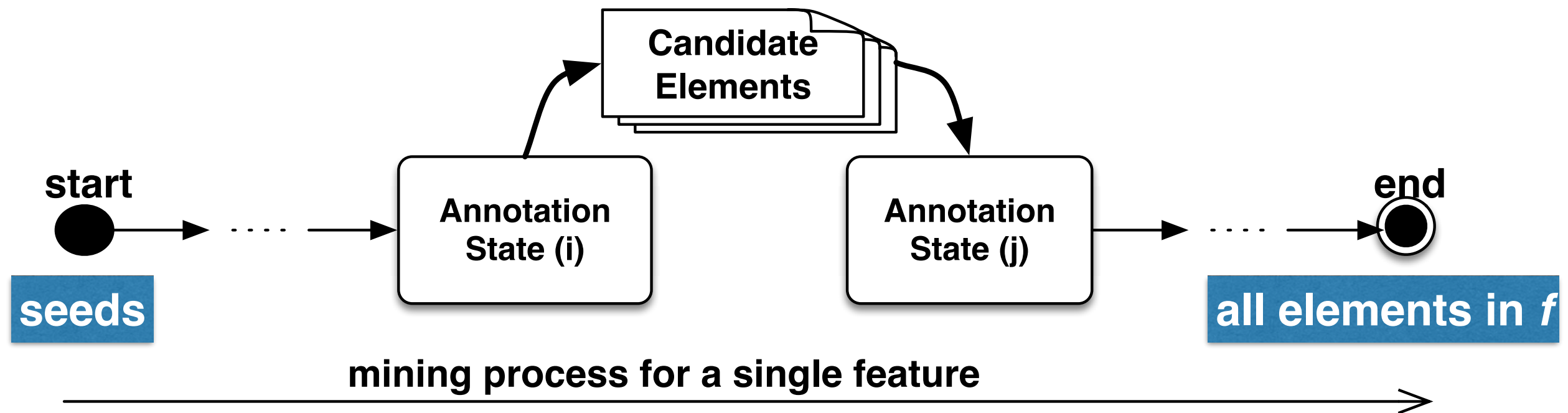
(e, f)

$(e, f) \mid (e, f) \in A, g \Rightarrow f$

Annotate Features

Annotation State

$$S(A^*, f, i) = \{e \mid (e, f) \in A^*\}$$



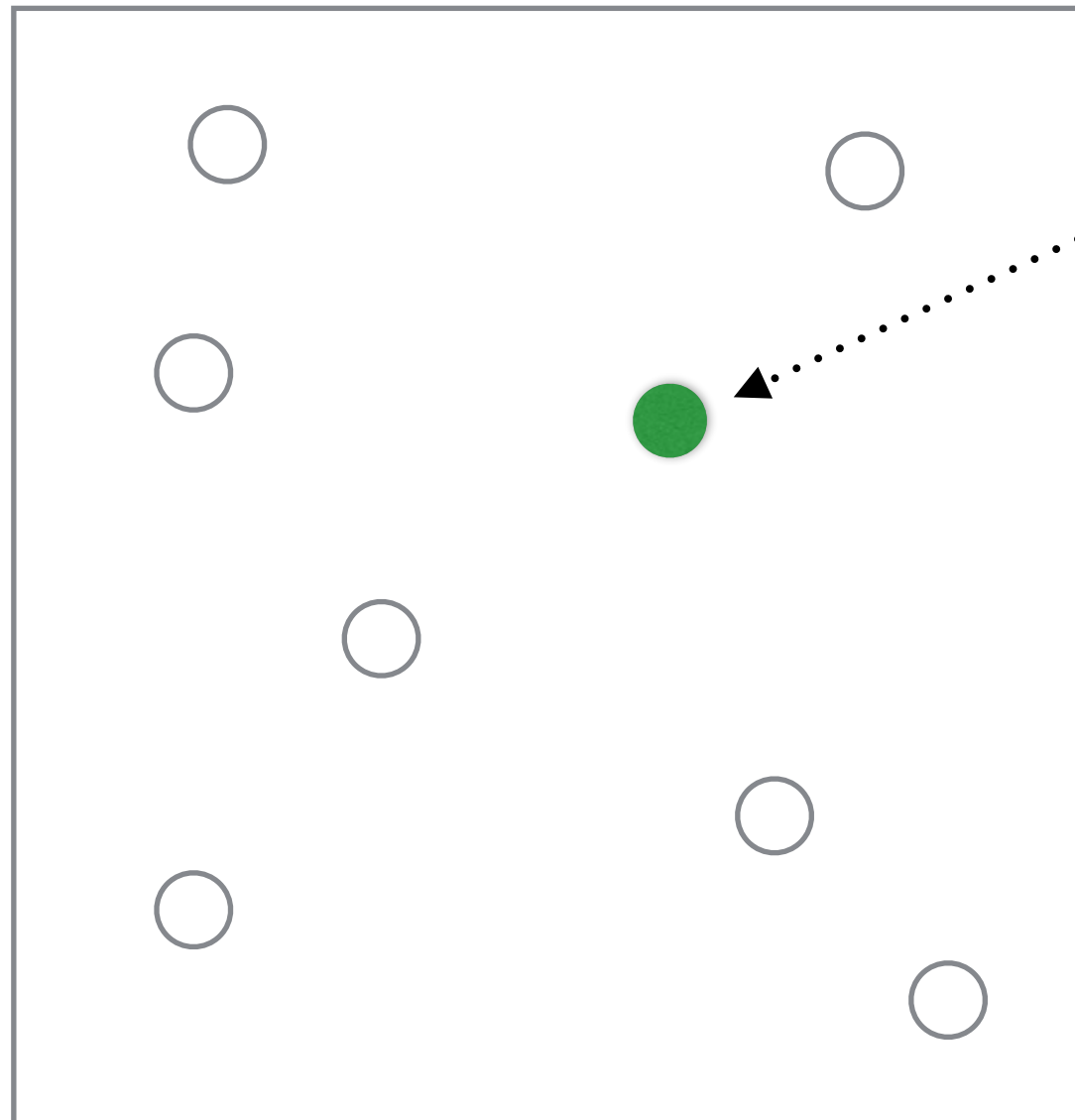
Feature-Element Correlation Coefficient

Prog. Elements Cand.

feature f

$$S(A^*, f, i)$$

Annotation
State (i)



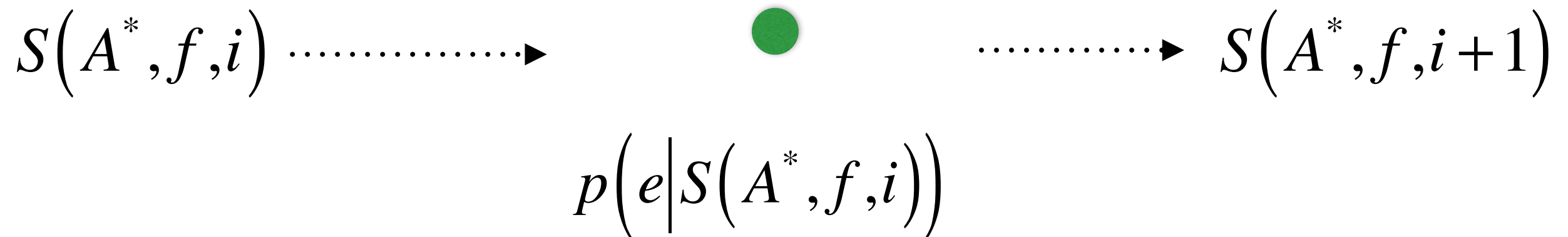
$$S(A^*, f, i+1)$$

Annotation
State (i+1)

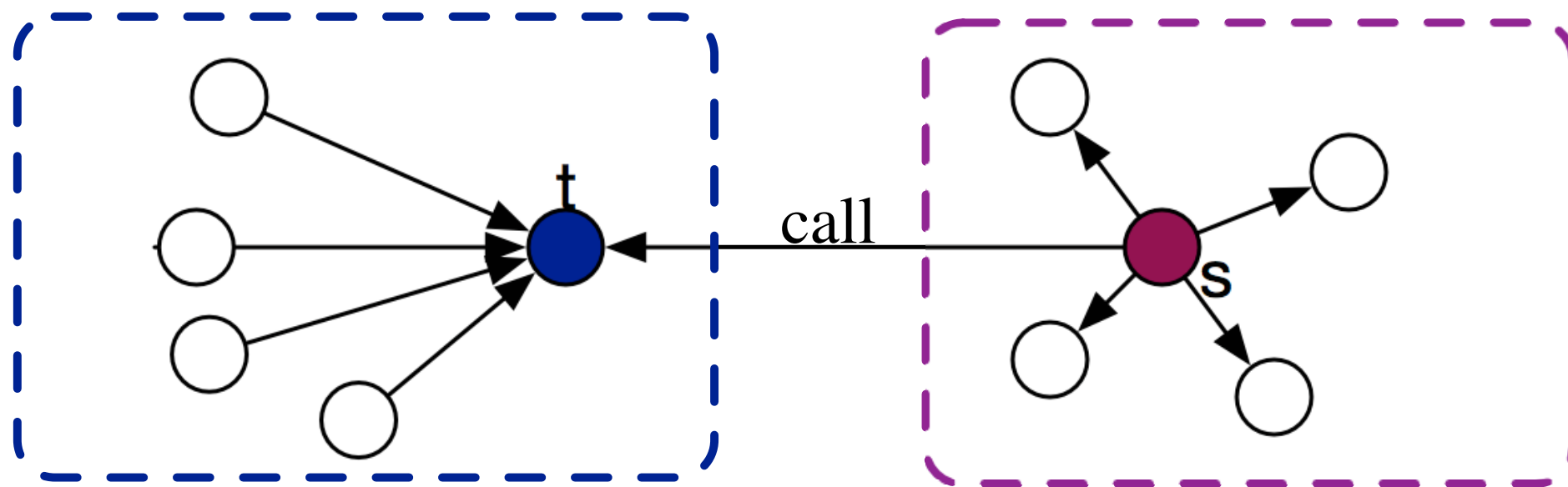
Feature-Element Correlation Coefficient(cond')

Prog. Elements Cand.

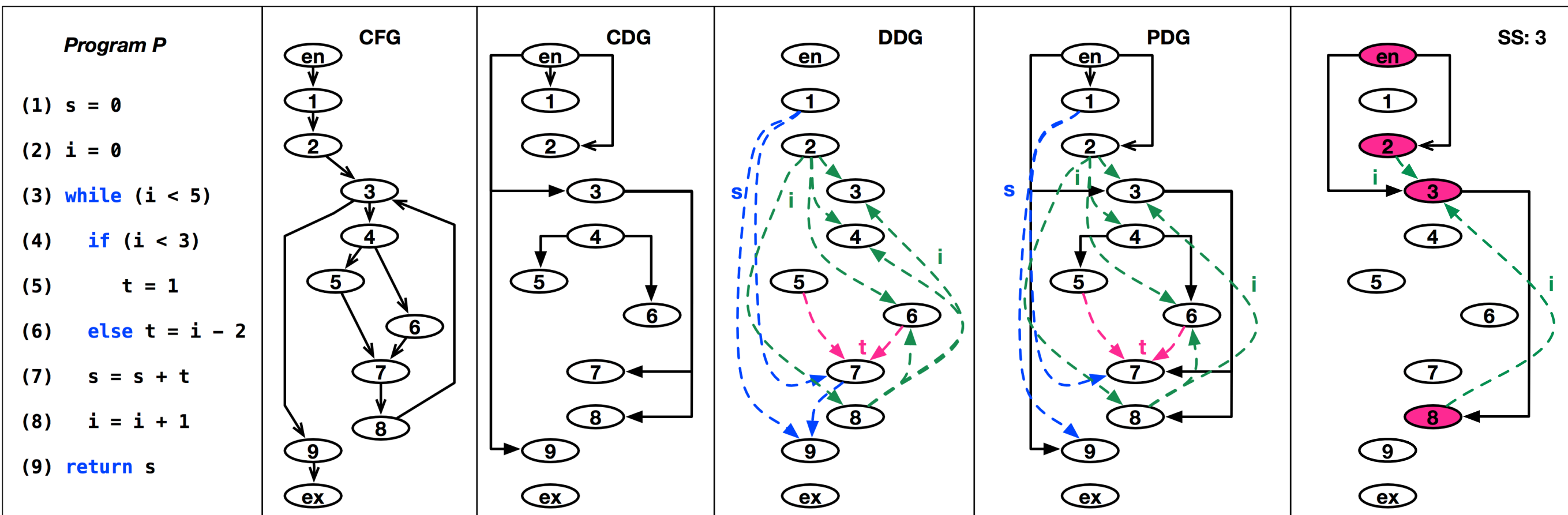
feature f



Feature-Element Correlation Coefficient(cond')



Feature-Element Correlation Coefficient(cond')



Slicing Scope

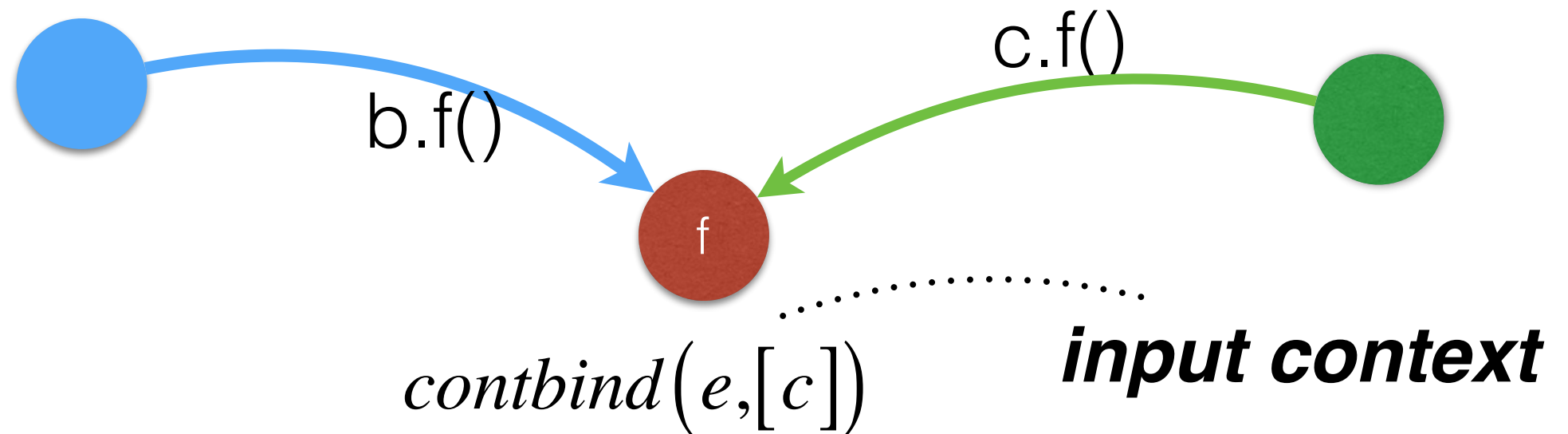
$$sscope(e) = e \cup \left\{ s \mid s \xrightarrow[cf]{df} e, s \in E \right\}$$

Feature-Element Correlation Coefficient(cond')

Binding

$$\textit{bind}(e) = \textit{def}(e) \cup \textit{use}(e)$$

Context Binding



Context Binding

Context Binding @ Method Invocation

callsite $l = r_0.m(r_1, r_2, \dots, r_n)$

$contbind(m, [r_1, \dots, r_n]) = dispatch(p_i = r_i) \rightarrow bind(m)$

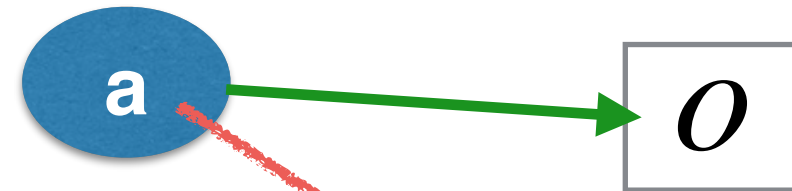
Context Binding

Context Binding @ Method Invocation

```
main(){
```

```
    A a = new A();
```

```
csite 1: z = wrapper(a);  
}
```



callsite 1

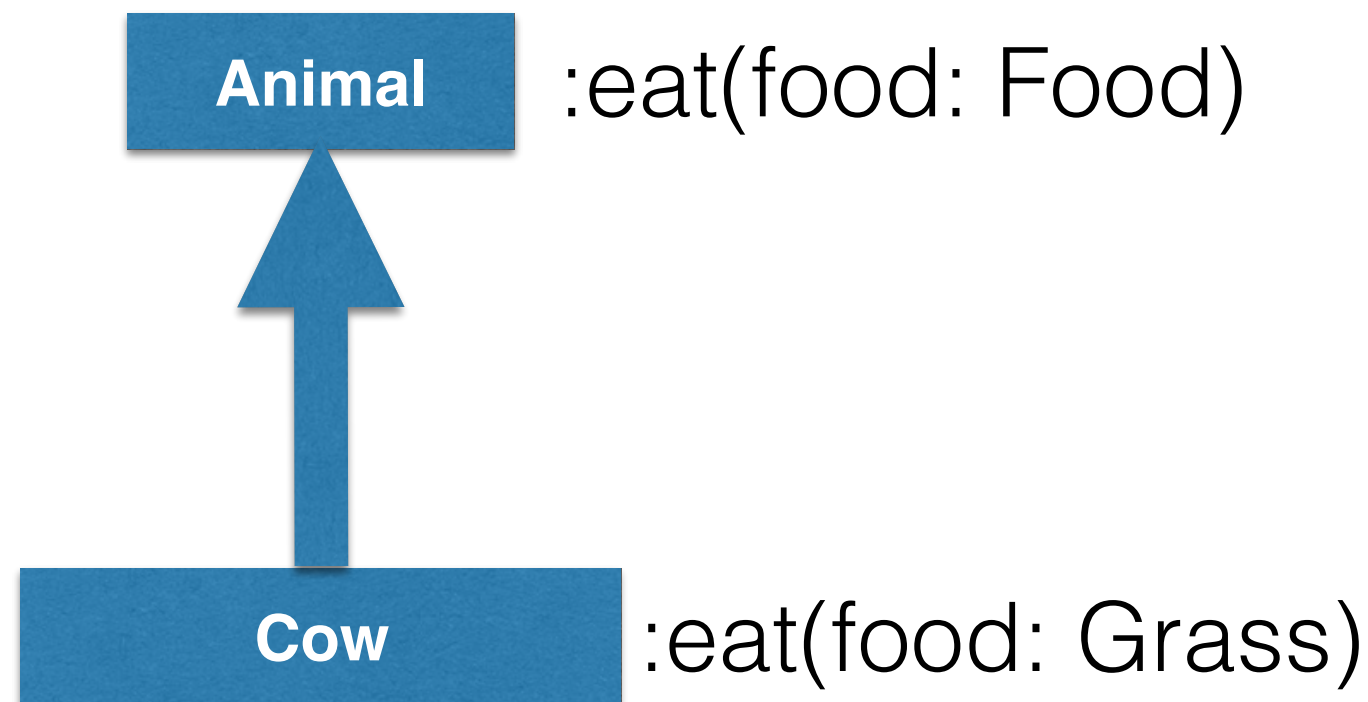
```
wrapper (A b){  
    y = bar(b); return y;  
}
```

```
bar (A c){  
    x = c.f; return x;  
}
```

contbind(wrapper(..),[a])

Context Binding

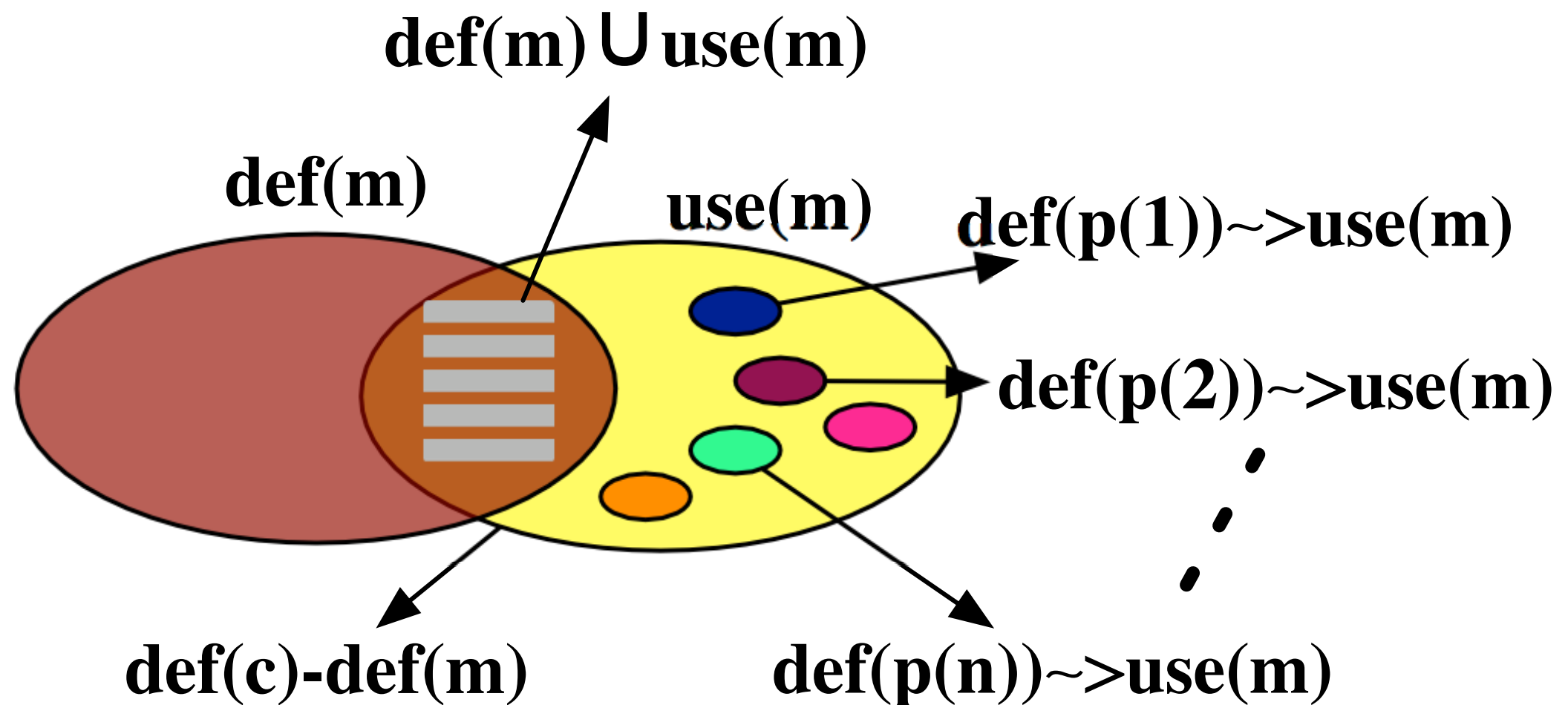
Context Binding @ Overriding



- 1. def and use in m
- 2. def in parent class/interface, used in m

Context Binding

Context Binding @ Overriding



Context Binding

Context Binding @ Overriding

defined in m : $def(m)$

used in m : $use(m)$

1. used in m and defined in m
2. used in m and not defined in m

$$contbind(m, [p_1, \dots, p_n]) = def(m) \cup \bigcup_{i=1}^n (use(m) \leadsto def(p_i))$$

\leadsto used to specify the source of the context

Context Binding

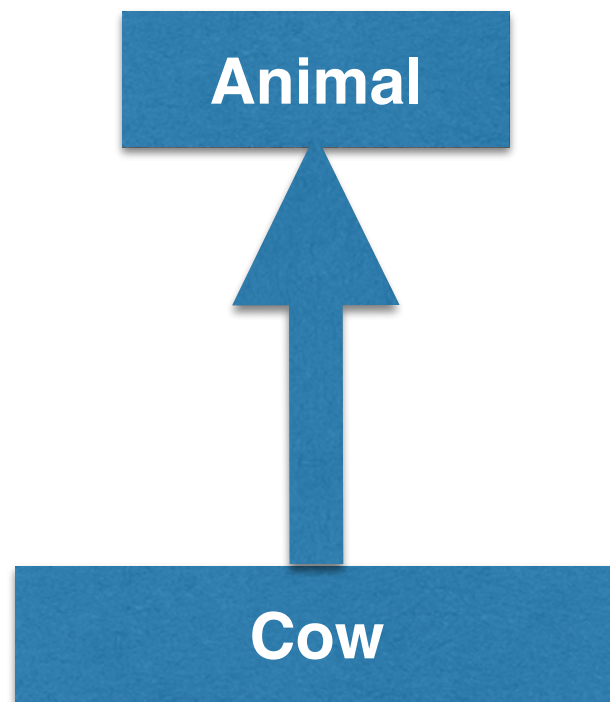
Context Binding @ Overriding : **Example**

```
1 public class FlyingCar implements OperateCar {  
2     public int startEngine(int encryptedValue) {  
3         OperateCar.super.startEngine(OperateCar.encryptedValue);  
4     }  
5 }  
-----  
6 public interface OperateCar {  
7     int encryptedValue = 1; ←  
8     default public int startEngine(int value) {...}  
9 }
```

contextbind(startEngine) = {encryptedValue, OperateCar.encryptedValue}

Context Binding

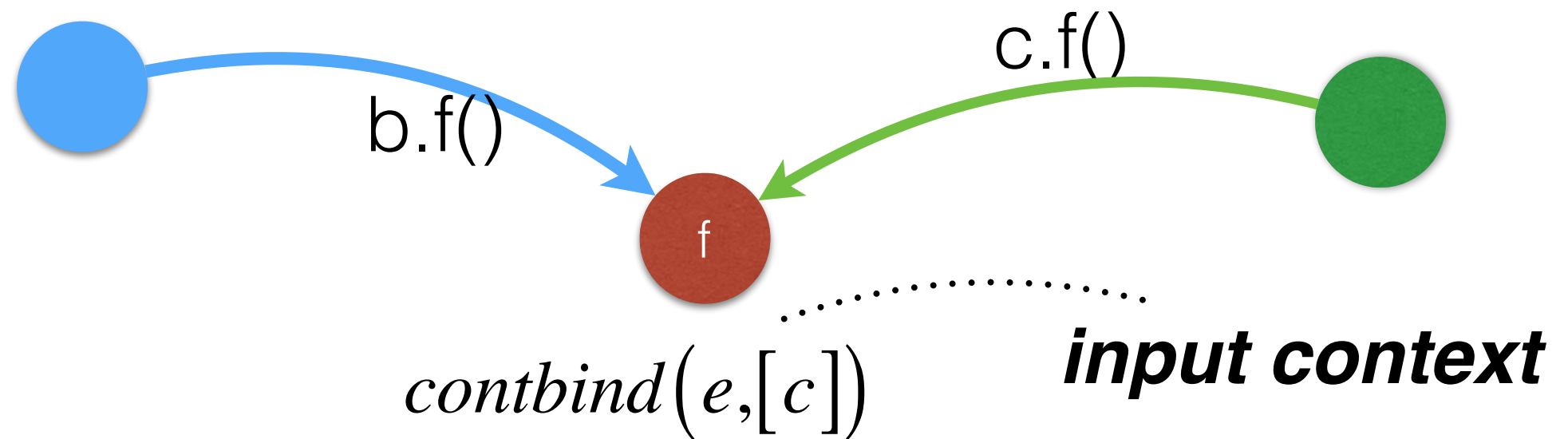
Context Binding @ Inheritance



$$\text{contbind}(c, [p_1, \dots, p_n]) = \text{bind}(c) \cup \bigcup_{i=1}^n \text{def}(p_i)$$

Feature-Element Correlation Coefficient(cond')

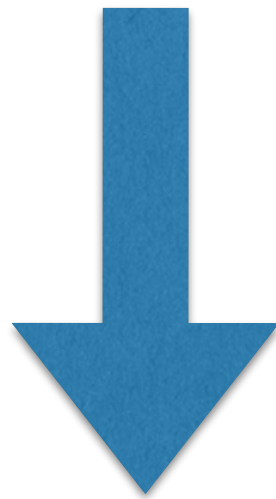
Context Binding



by default: ***context-aware points-to analysis***

Feature-Element Correlation Coefficient(cond')

$$S(A^*, f, i) = \{e \mid (e, f) \in A^*\}$$



$$S(A^*, f, i) = \bigcup_{a \in S(A^*, f, i)} \text{contextbind}(a)$$

StiCProb

1. Build Program DB
2. Build uniqueness table
3. Annotate features

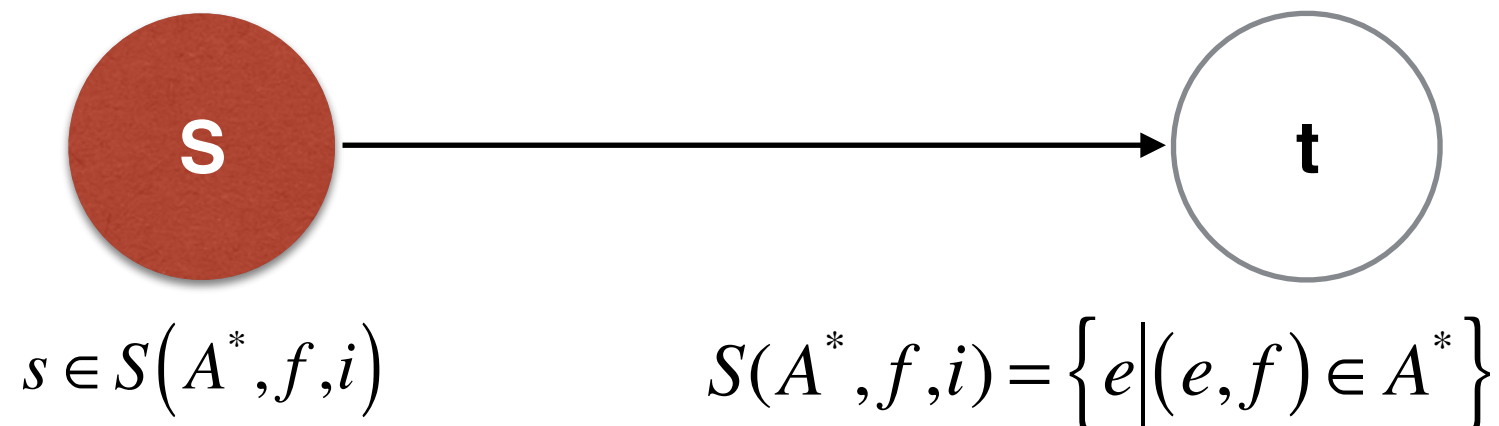
StiCProb: Uniquess Table

element s and t with a relation r $s \xrightarrow{r} t$

$$U\left(E, T, R, P_{forward}, P_{backward}\right)$$

$P_{forward}$

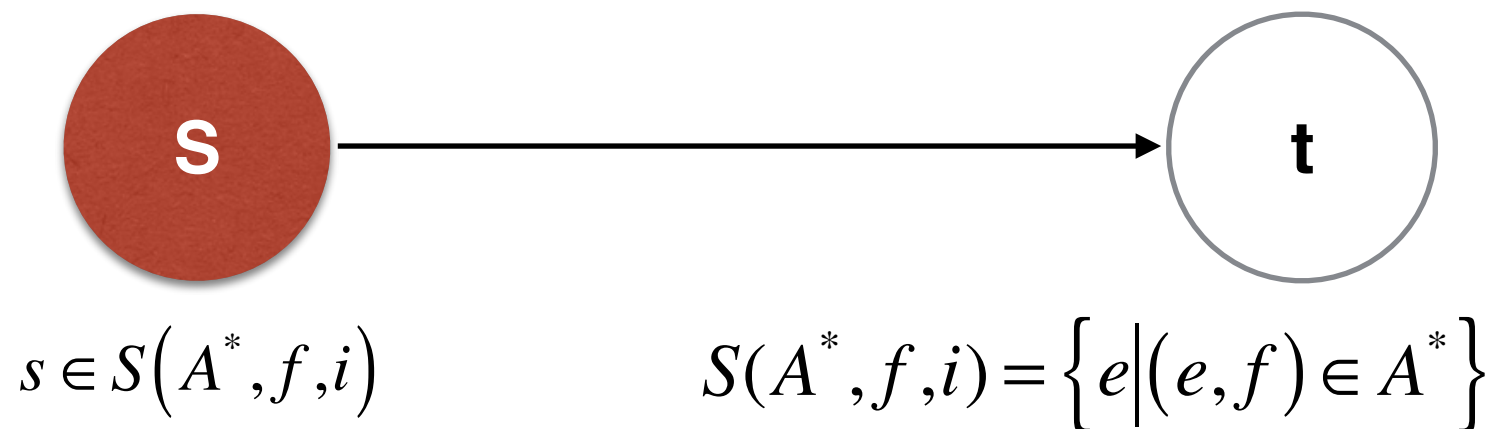
the uniqueness of t to s if s has been annotated to a feature **f**



StiCProb: Uniquess Table(cond')

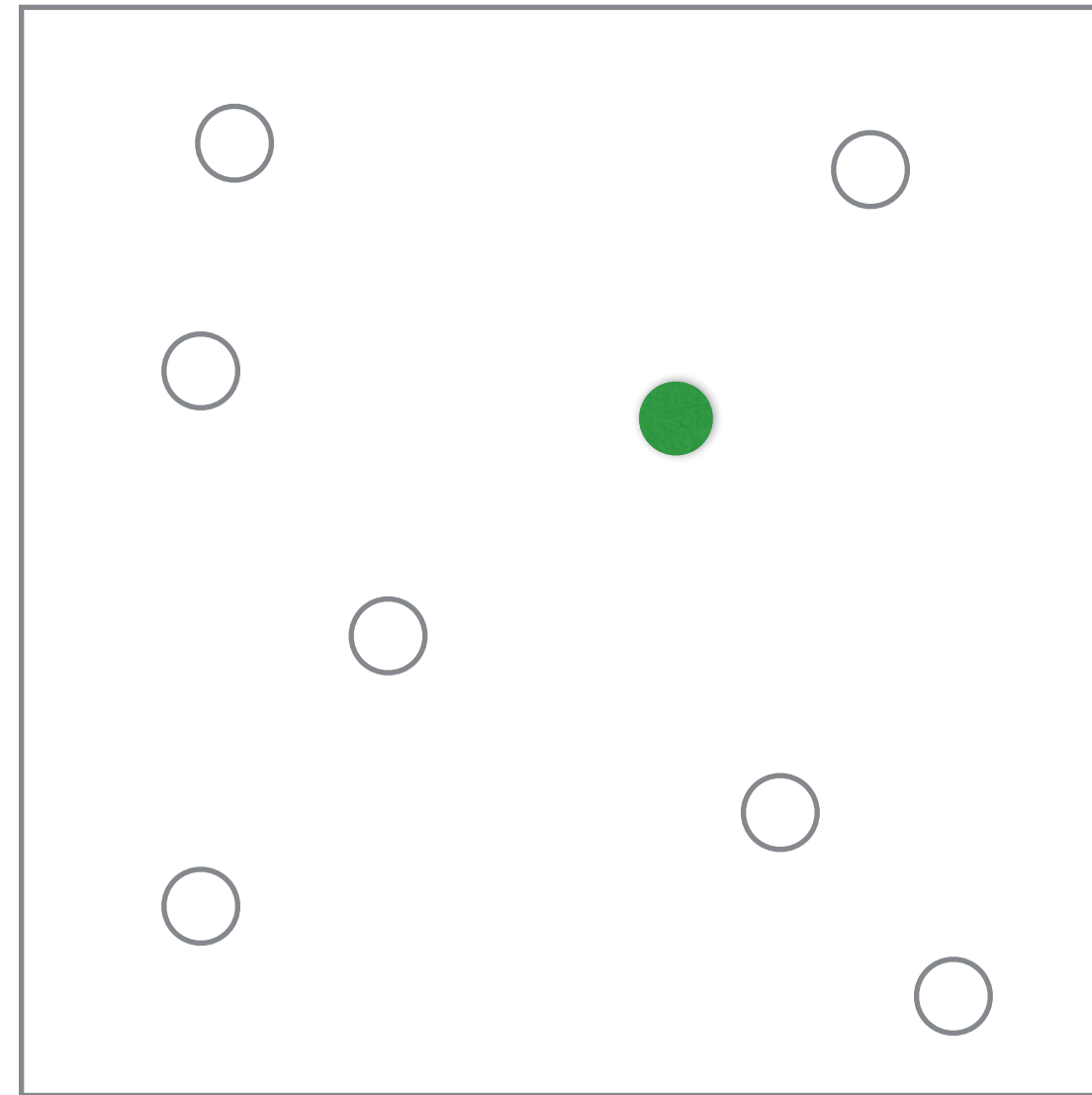
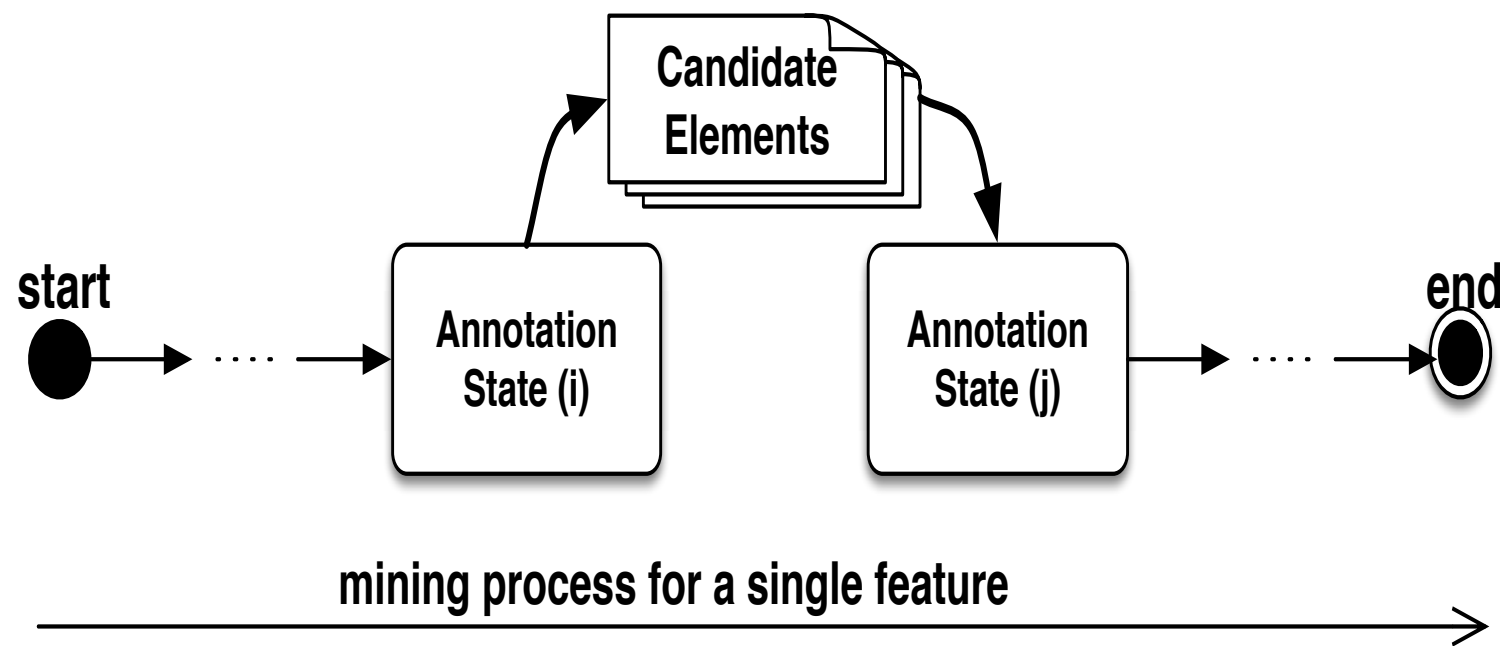
$P_{forward}$

the uniqueness of t to s if s has been annotated to a feature **f**



$$P_{forward} \left(s \xrightarrow{r} t \mid (s, f) \in A^* \right) = \frac{contbind(t, [s])}{contbind(s)}$$

StiCProb: Uniquess Table(cond') Cand.



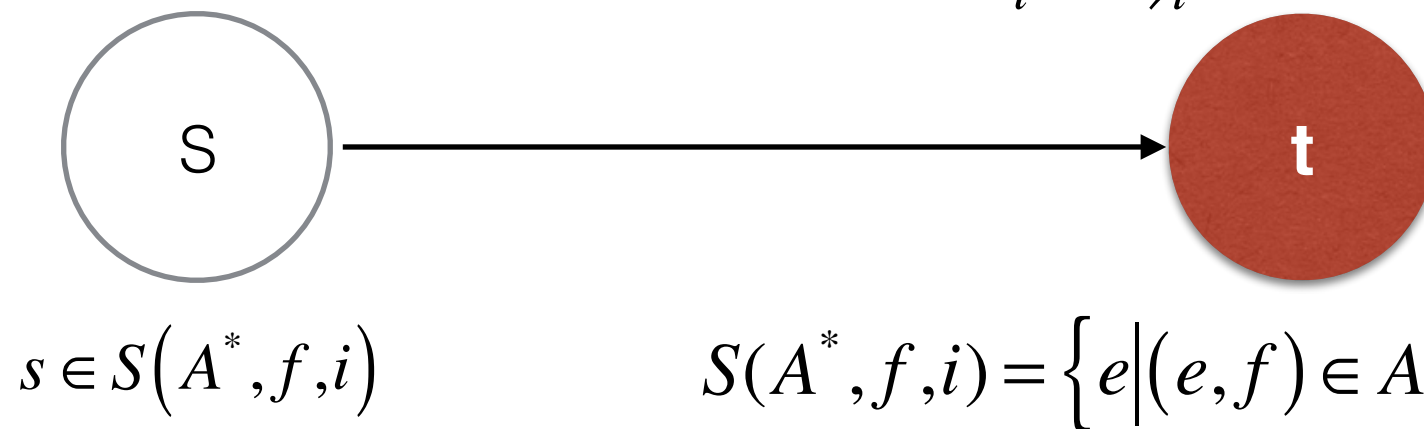
$$P_{forward} \left(s \rightarrow t \mid (s, f) \in A^* \right) = \frac{contbind(t, [s])}{contbind(s)}$$

StiCProb: Uniquess Table(cond')

$P_{backward}$

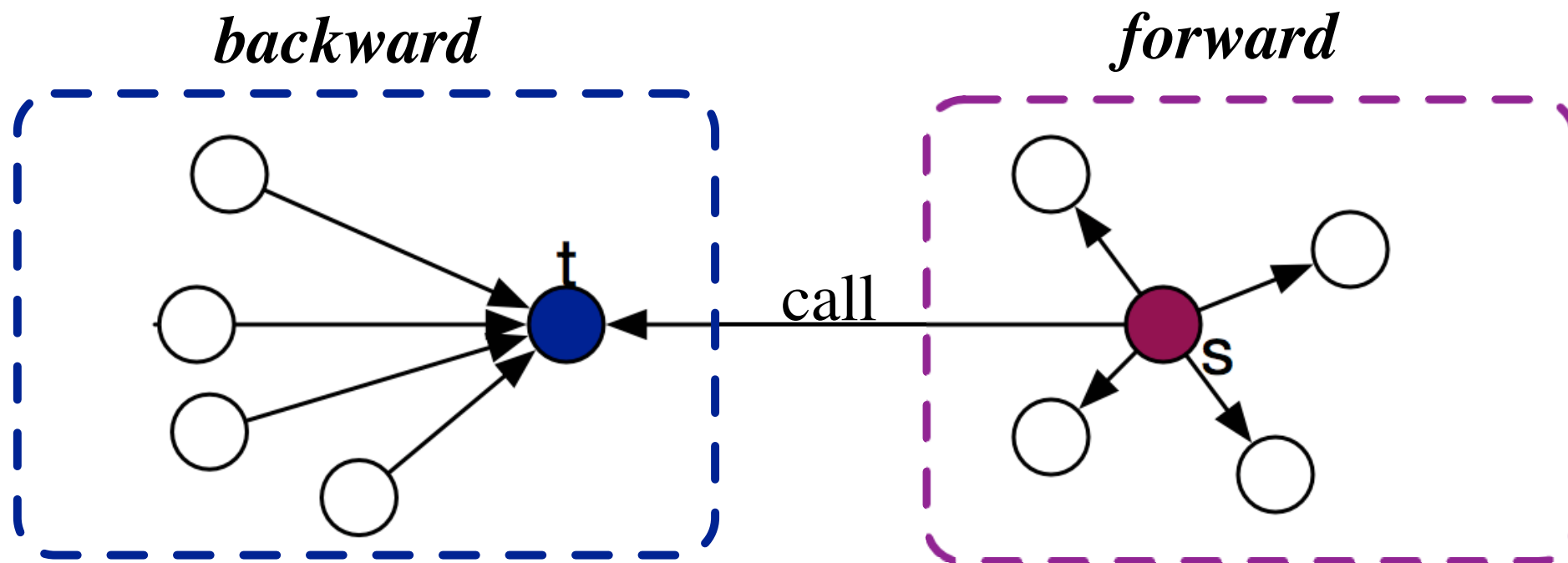
the uniqueness of s to t if t has been annotated to a feature **f**

$$P_{backward} \left(s \xrightarrow{r} t \mid (t, f) \in A^* \right) = \frac{contbind(t, [s])}{\bigcup_{i \xrightarrow{r} t} contbind(t, [i])}$$

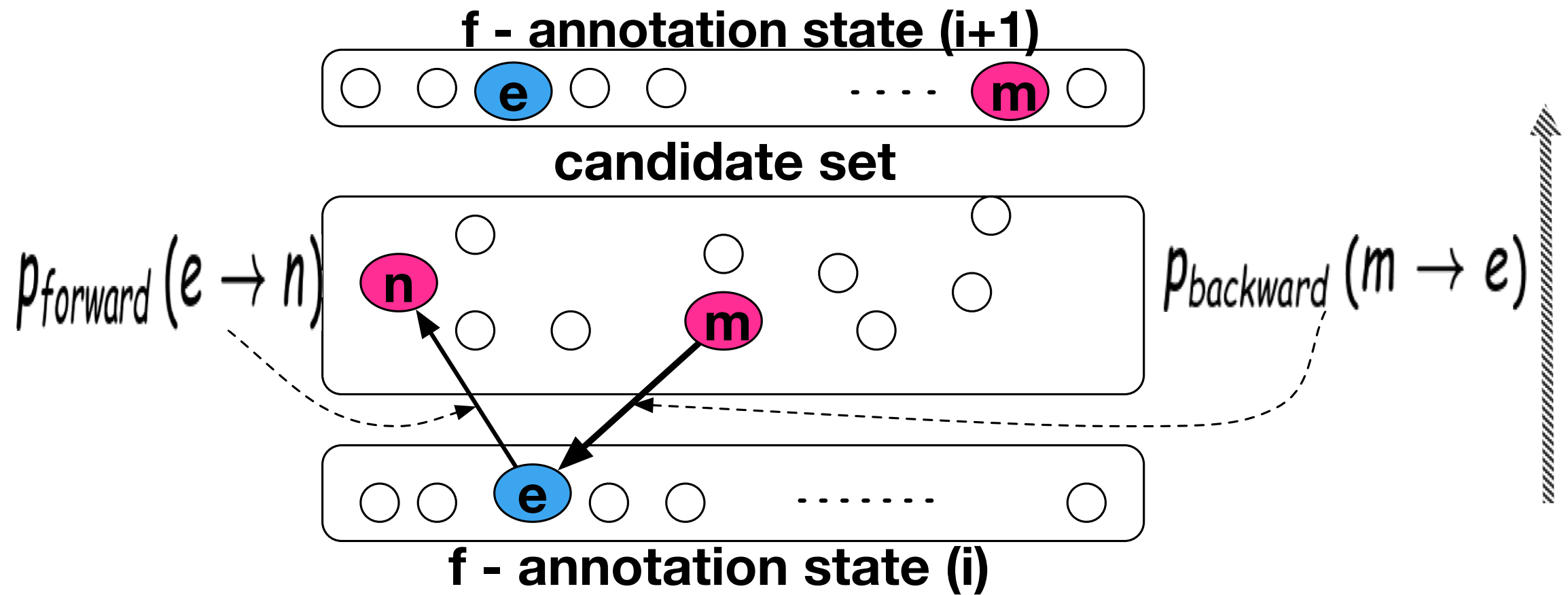


$\bigcup_{i \xrightarrow{r} t} contbind(t, [i])$ a collection of context binding from all prog. elements, which have relation r with t .

StiCProb: Uniquess Table(cond')



StiCProb



StiCProb

Algorithm 1: StiCProb feature mining approach

Input: $seeds, fm, threshold, U$

Output: all annotation states for features $Sset$ in fm

```
1 Create a set of annotation states as
   $Sset = \bigcup_f^{f \in features} S(A^*, f);$ 
2 Assign seeds to each feature as  $S(A^*, f) = seeds(f);$ 
3 Create feature set  $features$  with all features in  $fm$ ;
4 while  $features$  not NULL do
5   for feature  $f$  in  $features$  do
6     Create set  $waitList = \emptyset$ ;
7     Create candidate set  $C(S, f) = \emptyset$  for  $f$ ;
8     Add all elements have relations with elements in
        $S(A^*, f)$  to  $C(S, f)$ ; // initialize  $C(S, f)$ 
9     for element  $m$  in  $C(S, f)$  do
10      if there is a relation  $r$  from  $m$  to the element
         $e$  in  $S(A^*, f)$  then
11        Let  $value =$ 
           $p_{backward}(m \xrightarrow{r} e | (e, f) \in A^*);$ 
12      else
13        Let  $value =$ 
           $p_{forward}(e \xrightarrow{r} m | (e, f) \in A^*);$ 
14      if  $value > threshold$  then
15        Add  $m$  to  $waitList$ ;
16      Update  $S(A^*, f) \leftarrow S(A^*, f) \cup waitList$ ;
17      if  $StopCheck(f)$  is TRUE then
18        Remove  $f$  from  $features$ ;
19 return  $Sset$ ;
```

Feature model(fm)

Seeds(seeds)

threshold

Uniqueness Table(U)

Case Study

Projects	LOC	#features	domain
Prevalyer	8,009	5	object persistence library
MobileMedia	4,653	6	mobile
Lampiro	44,584	*2	message client
ArgoUML	~120K	7	modeling tool

Case Study

Experimental Setting:

seeds: FLAT3 tool

tool: Loong Eclipse plugin

feature model: benchmark

benchmark

Related Approaches:

Type system, Topology analysis, Text comparison

Measurement:

precision recall f-score

Case Study

Other Settings:

seeds: 3

threshold: 0.6

Experimental Result

StiCProb with threshold $t = 0.6$

Project	Feature	Feature Size			Mining Results		
		LOC	FR	FI	IT	Recall	Prec.
Prevayler	Censor	105	10	5	3	17%	60%
	Gzip	165	4	4	3	16%	100%
	Monitor	240	19	8	2	17%	82%
	Replication	1487	37	28	26	79%	98%
	Snapshot	263	29	5	9	42%	99%
MobileM.	CopyMedia	79	18	6	4	43%	95%
	Sorting	85	20	6	4	32%	100%
	Favorites	63	18	6	12	20%	100%
	SMS Trans.	714	26	14	23	91%	49%
	Music	709	38	16	4	39%	90%
	Photo	493	35	13	5	63%	61%

LOC: line of code, **FR:** count of distinct code fragments, **IT:** number of iteration,
Prec.: precision

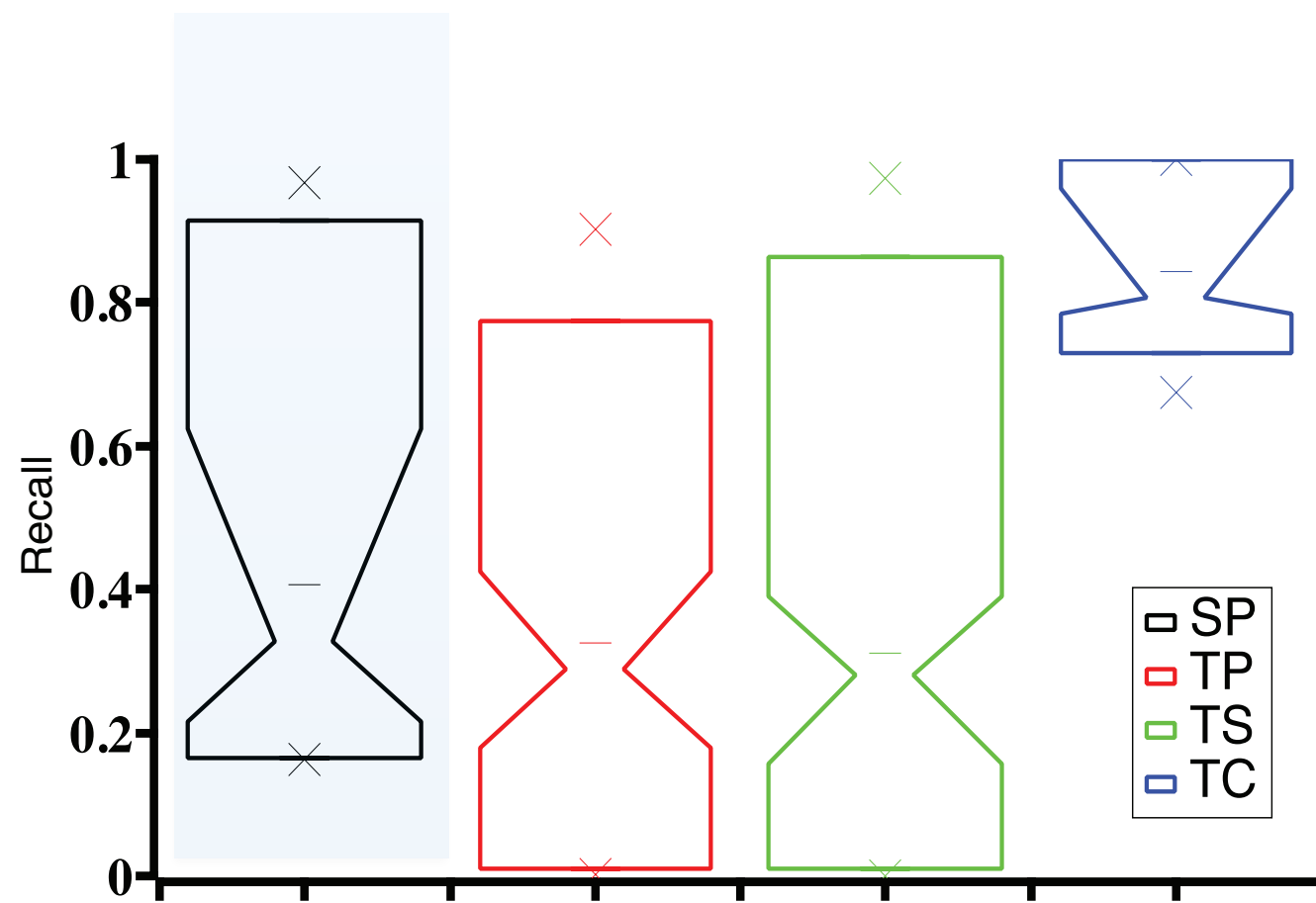
Experimental Result (cond')

StiCProb with threshold $t = 0.6$

Project	Feature	Feature Size			Mining Results		
		LOC	FR	FI	IT	Recall	Prec.
MobileM.	M.Transfer	153	4	3	14	97%	94%
Lampiro	Compre.	5155	33	20	34	40%	82%
ArgoUML	Cognitive	16319	285	233	127	70%	92%
	Activity	2282	115	80	17	26%	74%
	State	3917	115	88	18	33%	82%
	Collab.	1579	53	40	40	17%	72%
	Sequence	5379	65	53	98	33%	89%
	Use-Case	2712	59	49	39	19%	70%
	Deployment	3147	57	47	36	22%	67%

LOC: line of code, **FR:** count of distinct code fragments, **IT:** number of iteration,
Prec.: precision

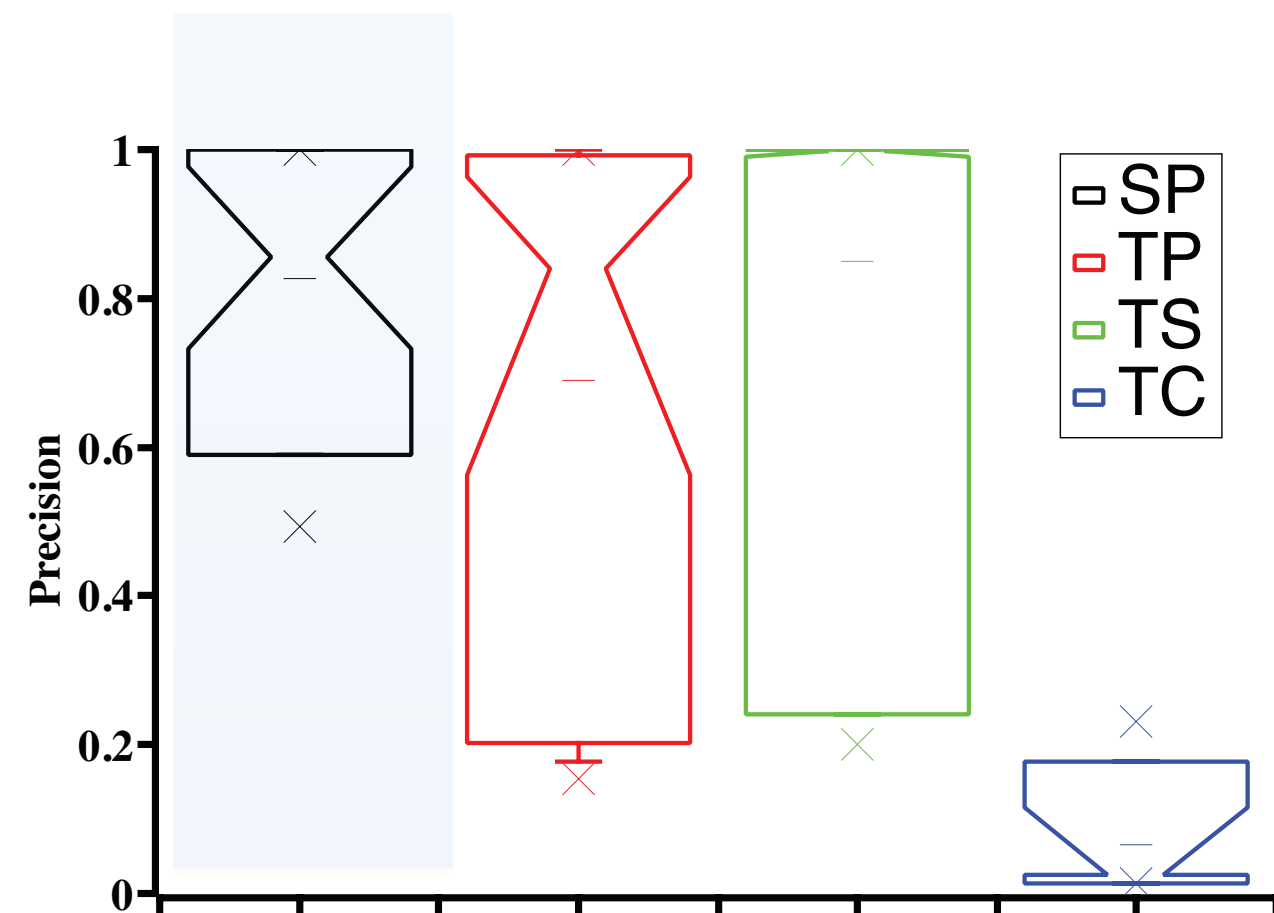
Experimental Result (cond')



Recall Performance

SP: StiCProb ($t = 0.6$)

TP: topology analysis



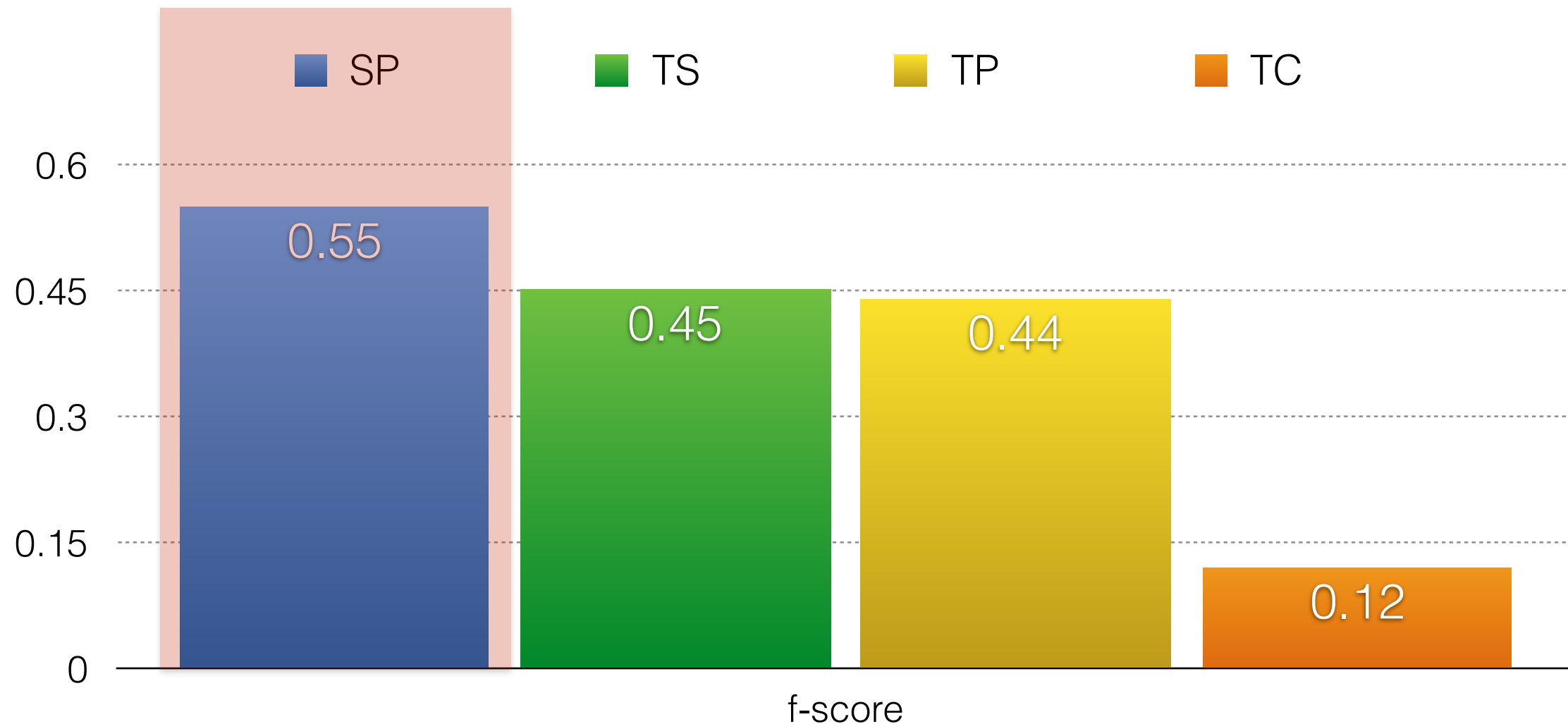
Precision Performance

TS: type system

TC: text comparison

Experimental Result

f-score



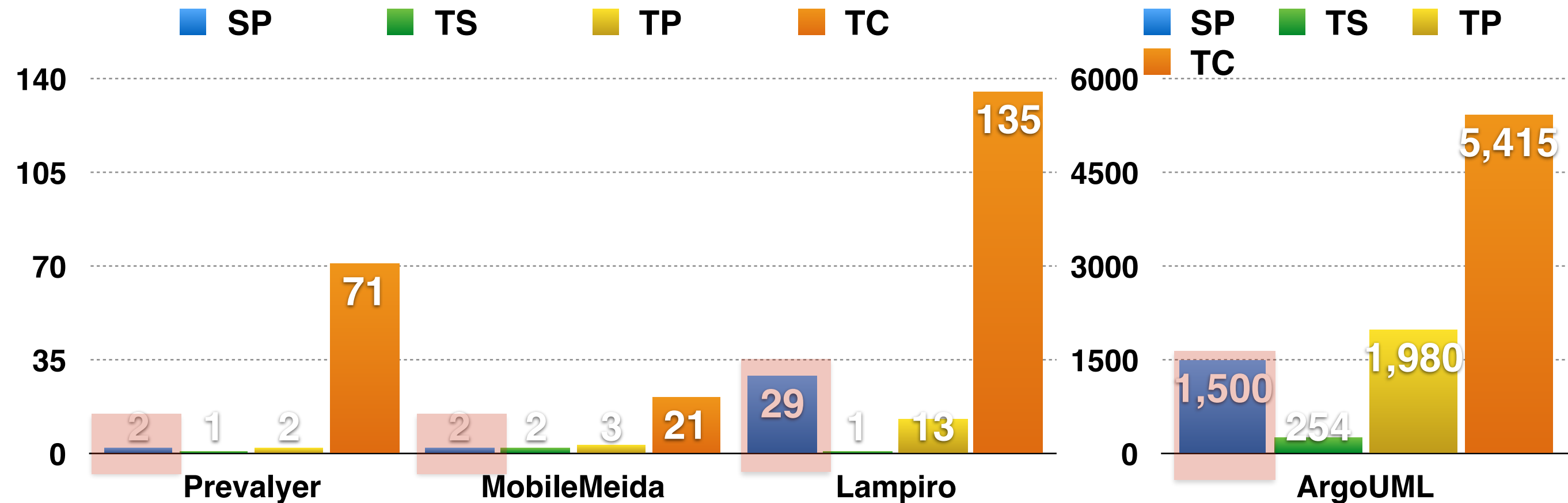
SP: StiCProb ($t = 0.6$)

TP: topology analysis

TS: type system

TC: text comparison

Experimental Result Runtime



SP: StiCProb ($t = 0.6$)

TP: topology analysis

TS: type system

TC: text comparison

Discussion

Seeds:

1. seeds provided by FLAT3 might be not correct
2. number of seeds
3. granularity of seeds: coarse granularity could improve the recall performance, but sometimes at the cost of precision.

Discussion

Thresholds:

threshold: 0.6 \longrightarrow 0.8

precision: 83% \longrightarrow 85%

The *threshold* contributes less to the performance.

Structure of the program

Thanks

Loong Plugin

- Download: <http://www.chrisyttang.org/loong/>
- Source code: <https://github.com/csyttang/Loong>
- Experimental results: <https://drive.google.com/folderview?id=0B9l0qvk6pnW0ZDRYMmxlQVhRb0U&usp=sharing>
- Online Tutorial: <http://www.chrisyttang.org/loong/>