DUET

상향식 나열 방법의 최적화와 확장 계획에 관하여

박준성 2023-10-09

DUET

목차

프로그램의 정수 표현식

DUET

프로그램의 정수 표현식 - 접근

상향식 나열 방법에서는 크기가 큰 프로그램을 합성하기 위해 크기가 작은 프로그램을 부품으로 활용한다.

기존의 duet은 하나의 프로그램을 표현하기 위해 expr라는 자료형을 사용한다.

다음은 expr으로 표현한 부품과 그로 합성한 프로그램을 예로 나타낸다.

```
ex. (str.substr arg0 0 (- (str.len arg0) 1))
-> (str.++ (str.substr arg0 0 (- (str.len arg0) 1)) '.')
```

DUET

프로그램의 정수 표현식 - 접근

예제를 통해 확인할 수 있는 것처럼,
크기가 큰 프로그램은 **크기가 작은 프로그램의 모든 부분을 그대로 포함**하고 있다.

크기가 커지면 커질 수록 부품으로 사용할 수 있는 프로그램의 개수는 많아지므로
크기가 큰 프로그램을 효율적으로 관리하지 못하면 메모리 공간을 많이 사용한다.

따라서, 프로그램을 표현할 때 중복된 부분을 최대한 가볍게 표현하고자
프로그램을 하나의 정수로 표현하는 방법에 대해서 생각해보았다.

DUET

프로그램의 정수 표현식 - 구현

프로그램 P와 일대일 대응되는 정수 값(이하, P의 인덱스)을 n(P)라고 하자.

f가 k항 연산자일 때, f로 합성한 프로그램 F는 다음과 같이 표현할 수 있다.

`F = (f, [n(sub_1); n(sub_2); … ; n(sub_k)])`

동일한 프로그램을 expr로 표현하면 다음과 같다.

`expr(F) = (f expr(sub_1) expr(sub_2) … expr(sub_k))`

DUET

프로그램의 정수 표현식 - 구현

정수 표현식은 다음과 같이 구현한다.

1. 크기가 1인 프로그램(단말기호)와 인덱스를 미리 매핑한다.

   ex. 0 → “.” ; 1 → “+” ; 2 → arg0 …

2. 크기가 1보다 큰 프로그램은 이전 슬라이드에서 설명한 것과 같이 매핑한다.

   ex. 3 → (str.++, [0;1]) = “.+” = (str.++ “.” “+”)

DUET

프로그램의 정수 표현식 - 통계

정수 표현식을 채택했을 때 보이는 성능은 다음과 같다.

Domain	Bottom-up	Solved/ Total	T		T_{BU}		Size	
			Avg.	Med.	Avg.	Med.	Avg.	Med.
String	rasauq1122/duet	204/205	2.12	0.01	0.04	< 0.01	27.4	13
	wslee/duet	204/205	3.55	0.01	1.43	< 0.01	27.3	13
Bitvec	rasauq1122/duet	645/750	20.2	5.08	2.17	0.08	350.9	46
	wslee/duet	645/750	21.9	5.16	3.42	0.18	350.9	46

표현식 평가의 최적화

DUET

표현식 평가의 최적화 - 접근

상향식 나열 방법에서 중복된 프로그램을 가지치기(pruning)하기 위해서
명세 조건으로 주어진 예제에 대한 프로그램의 출력을 확인하는 과정 또한
정수 표현식과 비슷한 원리로 최적화할 수 있다.

DUET

표현식 평가의 최적화 - 접근

기존 duet은 표현식을 평가할 때 param_valuation이라는
입력 예제로 주어진 인자들을 몇번째 인자인지에 대해서 묶는 작업을 진행한다.

```
ex. spec_input = [[a0; a1; a2]; [b0; b1; b2]; …; [z0; z1; z2]]
       param_valuation = [0 -> [a0; b0; …; z0];
                          1 -> [a1; b1; …; z1];
                          2 -> [a2; b2; …; z2]]
```

DUET

표현식 평가의 최적화 - 접근

예제의 개수가 N이고 합성할 프로그램의 인자의 수가 M이라고 하면,
`param_valuation`을 구하는 과정의 시간복잡도는 $\mathcal{O}(NM)$이다.

이미 만들어진 부품들의 예제에 대한 출력을 저장하여
위 과정의 시간복잡도를 $\mathcal{O}(M)$으로 줄이는 방법에 대해 소개한다.

DUET

표현식 평가의 최적화 - 구현

```
P = (str.substr (str.++ '.' arg0) 0 (- (str.len arg0) 1))
```

프로그램 P를 평가한 값은 P를 이루는 각 하위 프로그램을 먼저 평가한 후,
그 값을 P의 연산자(위의 경우, `str.substr`)로 평가한 값이다.

DUET

표현식 평가의 최적화 - 구현

```
P = (str.substr (str.++ ‘.’ arg0) 0 (- (str.len arg0) 1))
```

예제 입력이 [[“HYU”];[“PROGRAM”];[“SYNTHESIS”]]라고 하자.
그럼 각 하위 프로그램은 다음과 같이 평가된다.

```
   (str.++ ‘.’ arg0) -> [“.HYU”; “.PROGRAM’; “.SYNTHESIS”]
                   0 -> [     0;           0;             0]
(- (str.len arg0) 1) -> [     2;           6;             8]
```

DUET

표현식 평가의 최적화 - 구현

예제 입력이 [[“HYU”];[“PROGRAM”];[“SYNTHESIS”]]일때 프로그램 P의 출력은
예제 입력이 spec_in일 때의 프로그램 Q의 출력과 같다.

```
spec_in = [[“.HYU”; 0; 2]; [“.PROGRAM”; 0; 6]; [“.SYNTHESIS”; 0; 8]]
      Q = (str.substr arg0 arg1 arg2)
```

이때 param_valuation은 아래와 같고, 이는 하위 프로그램의 출력으로 쉽게 구할 수 있다.

```
param_valuation = [0 -> [“.HYU”; “.PROGRAM’; “.SYNTHESIS”];
                   1 -> [0; 0; 0];          2 -> [2; 6; 8]]
```

DUET

표현식 평가의 최적화 - 구현

f가 k항 연산자일 때, f로 합성한 프로그램 F는 다음과 같이 표현할 수 있다.

`F = (f, [n(sub_1); n(sub_2); … ; n(sub_k)]`

F를 빠르게 평가하기 위해서 변형시킨 후(이하, F′), 다음과 같은 expr로 표현한다.

`expr(F′) = (f arg0 arg1 … arg{k-1})`

F′가 F와 동치가 되도록 다음과 같이 param_valuation을 설정한다.

`[0 -> out(sub_1); 1 -> out(sub_2); …; k-1 -> out(sub_k)]`

DUET

표현식 평가의 최적화 - 구현

각 부품의 출력은 부품의 인덱스에 일대일 대응되므로 빠르게 구할 수 있다.

또한, 하위 프로그램의 출력을 평가하지 않고 바로 구할 수 있으므로
그에 따른 성능 향상도 기대해볼 수 있다.

DUET

표현식 평가의 최적화 - 통계

입력의 예제가 많아져도 T_{BU}의 상승폭이 크지 않는 걸 확인할 수 있다.

Bottom-up	bitvec/1_10.sl (max_component_size : 3)		bitvec/1_100.sl (max_component_size : 4)		bitvec/1_1000.sl (max_component_size : 4)	
	T	T_{BU}	T	T_{BU}	T	T_{BU}
rasauq1122/duet-1	7.11	0.04	70.72	0.41	240.81	25.66
rasaqu1122/duet-2	7.13	0.03	71.23	0.15	214.88	1.08

DUET

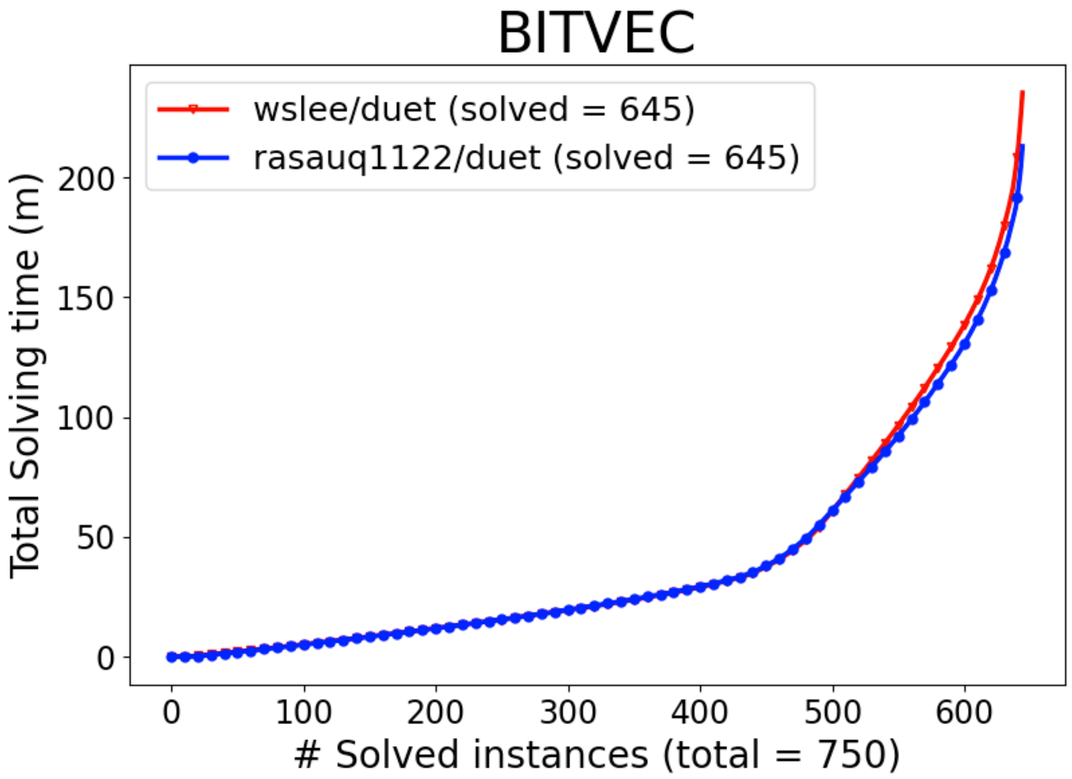
표현식 평가의 최적화 - 통계

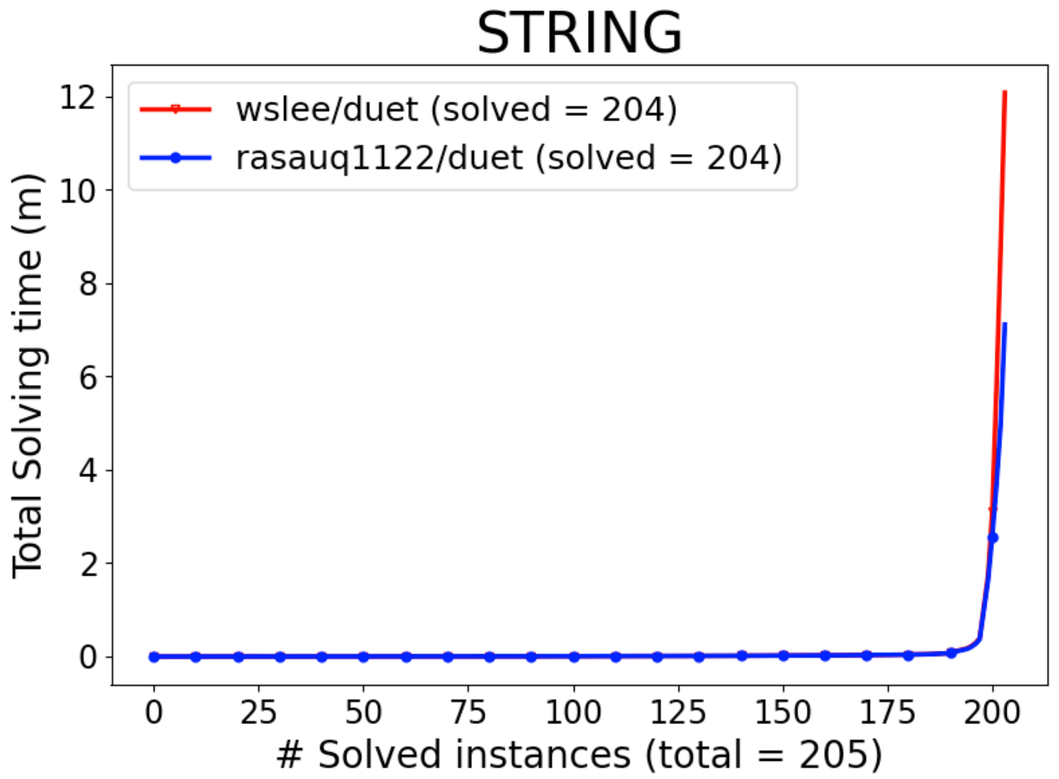
표현식 평가의 최적화를 이용했을 때 보이는 성능은 다음과 같다.

Domain	Bottom-up	Solved/ Total	T		T_{BU}		Size	
			Avg.	Med.	Avg.	Med.	Avg.	Med.
String	rasauq1122/duet	204/205	2.09	0.01	0.04	< 0.01	27.4	13
	wslee/duet	204/205	3.55	0.01	1.43	< 0.01	27.3	13
Bitvec	rasauq1122/duet	645/750	19.8	5.26	0.14	0.05	350.9	46
	wslee/duet	645/750	21.9	5.16	3.42	0.18	350.9	46

DUET

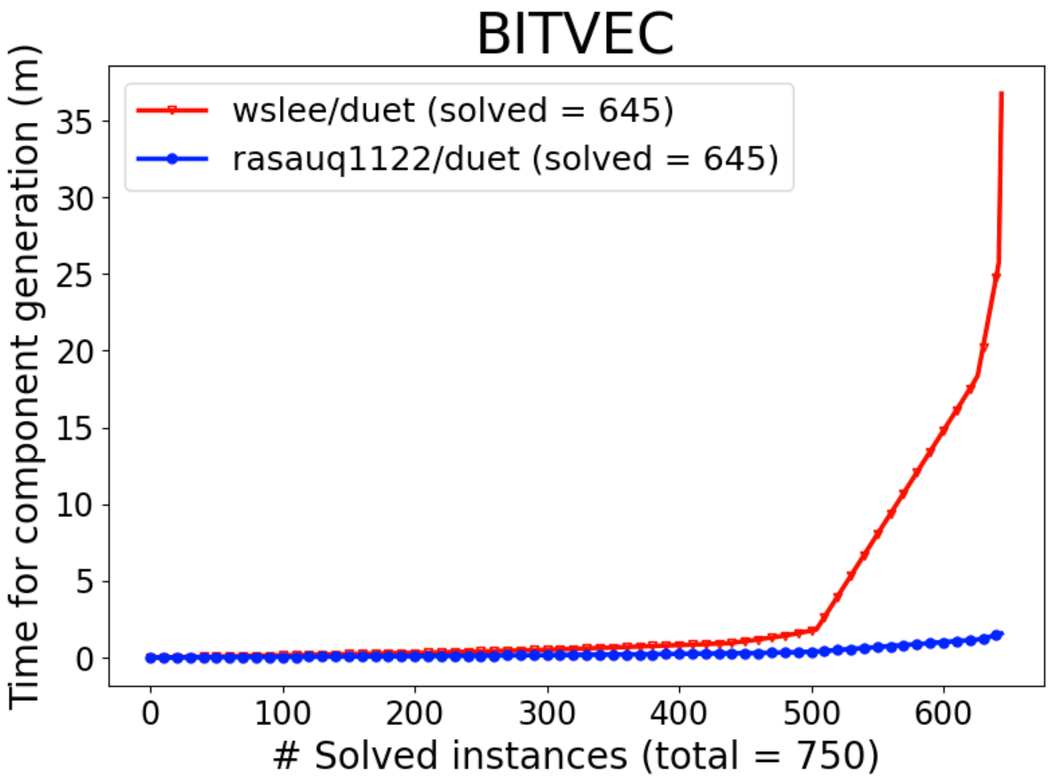
표현식 평가의 최적화 - 통계

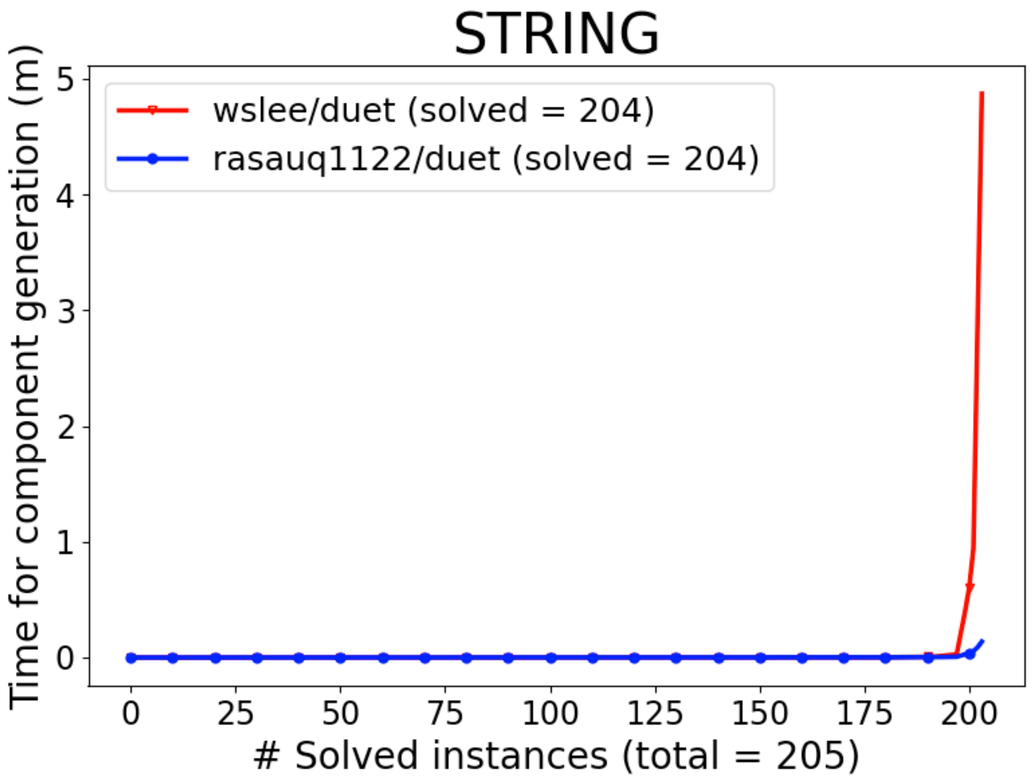




DUET

표현식 평가의 최적화 - 통계





DUET

표현식 평가의 최적화 - 통계

Domain	Bottom-up	T		T_{BU}		Size	
		Avg.	Med.	Avg.	Med.	Avg.	Med.
String	rasauq1122/duet-1	2.12	0.01	0.04	< 0.01	27.4	13
	rasauq1122/duet-2	2.09	0.01	0.04	< 0.01	27.4	13
Bitvec	rasauq1122/duet-1	20.2	5.08	2.17	0.08	350.9	46
	rasauq1122/duet-2	19.8	5.26	0.14	0.05	350.9	46

향후 확장 계획

DUET

향후 확장 계획

기존의 duet은 명세 조건이 programming by examples (PBE)이거나 oracle spec인 경우에만 프로그램을 합성할 수 있었다.

여기서 추가로 SMT Solver를 활용하여 논리식으로 주어진 명세 조건에서 입출력 예제를 얻어 프로그램을 합성할 수 있도록 확장할 계획이다.

DUET

향후 확장 계획

논리식 B가 항상 참이 되도록 하는 프로그램 f의 후보를 f′라고 하자.

다음과 같이 SMT Solver에 질의를 보낸다.

‘not B(f′)’이 충족가능(Satisfiable, SAT)한가?

UNSAT이라면 B(f′)가 항상 참이므로, f′ = f 다.
SAT이라면 B(f′)가 거짓이 되게 하는 입력 조건 I를 SMT Solver가 알려준다.

이제 f(I)가 어떤 값일 때 B가 참이 되는지 알 수 있다면 입출력 예제로 사용할 수 있다.

DUET

향후 확장 계획

f(I)의 값을 알기 위해 f(I)가 임의의 상수 c라고 가정하자.

‘(B(c) and x=I)’가 충족가능한가?

이는 항상 SAT이므로, 위 명제가 참이 되는 c의 값을 SMT Solver가 알려준다.

이로 인해 f(I) = c라는 입출력 예제를 얻을 수 있고,
이를 명세 조건에 추가하여 다시 프로그램 f의 후보 f′를 찾는다.

DUET

향후 확장 계획

예를 들어, 논리식 B와 f′이 다음과 같다고 하자.

$$B(f) = f(x,y) \geq x \wedge f(x,y) \geq y \wedge (f(x,y) = x \vee f(x,y) = y)$$
$$f'(x,y) = y$$

그럼 B(f′)은 다음과 같이 정리된다.

$$B(f') = y \geq x$$

x=1, y=0 일 때 not B(f′)는 참이므로, SMT Solver는 x=1, y=0을 I로 준다.

DUET

향후 확장 계획

`f(I) = f(1, 0) =` c로 가정하고, 다음과 같은 질의를 SMT Solver에게 보낸다.

$$B(c) \wedge x = I \equiv c \geq 1 \wedge c \geq 0 \wedge (c = 1 \vee c = 0)$$

c = 1일 때 (`B(c) and x=I`)는 참이므로, SMT Solver는 `f(1, 0) = 1`을 준다.

명세 조건에 새로운 입출력 예제를 추가하고 다시 프로그램 후보 f′를 합성한다.

DUET

향후 확장 계획

그러나 이 과정으로 항상 합성에 도움이 되는 입출력 예제를 얻을 수 있는 것은 아니다.

예를 들어, 논리식 B와 f′이 다음과 같다고 하자.

$$B(f) = (f(x) = f(x+3)) \wedge f(1) = 0 \wedge f(2) = 2$$

$$f'(x) = 2x - 2$$

$$B(f') = (2x - 2 = 2x + 5)$$

이 경우 x가 어떤 값이어도 not B(f′)는 참이다. (입력의 무용성)

DUET

향후 확장 계획

이전 슬라이드의 예에서 얻은 입력 조건 `I`는 x = 0이라고 하자.
그럼 `f(x) = f(0) = c, f(x+3) = f(3) = d`라고 가정하고, 다음과 같이 질의한다.

$$B(c,d) \wedge x = I \equiv (c = d)$$

c와 d가 서로 같기만 하면, 어떤 값이어도 (B(c,d) and x=I)는 참이다. (출력의 무용성)

DUET

향후 확장 계획

도움이 되지 않는 입출력 예제에 의해
명세 조건 자체가 참을 만족할 수 없게 되는 경우가 생길 수 있다.

그런 경우에는 해당 입출력 예제를 가질 수 없도록 명세 조건에 추가하고
처음부터 프로그램 합성을 시작한다.

DUET

향후 확장 계획

앞서 소개한 방법으로 논리식으로 주어진 명세 조건에 대해서도
프로그램 합성이 가능하도록 DUET을 확장할 계획이다.

일단은 linear integer arithmetic(LIA) 도메인이면서
사용자 정의 함수가 포함되지 않은 명세 조건인 경우에만 지원할 계획이다.