



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS

Tarea 2

INTEGRANTES

Torres Valencia Kevin Jair - 318331818
Aguilera Moreno Adrián - 421005200
Rivera Silva Marco Antonio - 318183583

PROFESORA

Karla Ramírez Pulido

AYUDANTES

Alan Alexis Martínez López
Manuel Ignacio Castillo López
Alejandra Cervera Taboada

ASIGNATURA

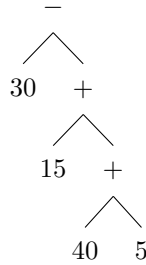
Lenguajes de Programación

9 de septiembre de 2022

1. Dadas las expresiones de WAE en sintaxis concreta, da su respectiva representación en sintaxis abstracta por medio de los Árboles de Sintaxis Abstracta correspondientes. En caso de no poder generar el árbol, justifica.

a) $\{-30 \{+15 \{+40\ 5\}\}\}$

La representación del ASA para la expresión anterior, tiene la forma siguiente

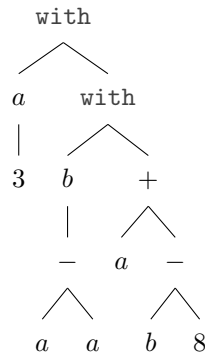


b) $\{-30 \{+300 \{+40\}\}\}$

En este caso, particular, las operaciones¹ no tienen sumandos y minuendo izquierdo. Por tanto, no podemos construir el ASA, pues sería un árbol incompleto (la expresión en sintaxis concreta es incompleta).

c) $\{\text{with } \{a\ 3\}$
 $\quad \{\text{with } \{b\ \{-a\ a\}\}$
 $\quad \{+a\ \{-b\ 8\}\}\}$

La representación del ASA para la anterior expresión es

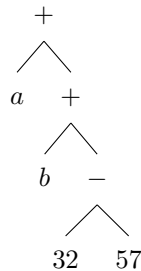


2. Dadas las siguientes expresiones de WAE en sintaxis concreta, obtén su sintaxis abstracta correspondiente y realiza la sustitución que se indica.

a) $e = \{+a \{+b \{-32\ 57\}\}\}$

(subst (parse e) 'a (add (num 3) (num 4)))

Primero demos la representación abstracta de la expresión anterior, esta es



¹2 sumas y una resta.

Luego, apliquemos la instrucción parse a la expresión e, esto es

```
(parse e) = (add (parse(a)) (parse({+ b {- 35 57} })))
           = (add (id 'a) (add (parse(b)) (parse({- 35 57} })))
           = (add (id 'a) (add (id 'b) (sub (parse(35)) (parse(57)))))
           = (add (id 'a) (add (id 'b) (sub (num 35) (num 57))))
           = e'
```

Ahora, apliquemos la instrucción subst a e' y los valores indicados, esto es

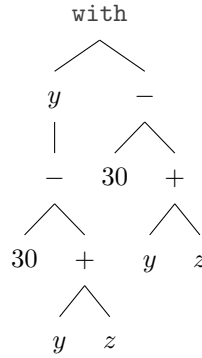
Obs. Por simplicidad, asumimos que

```
subst(e') = (subst (e') 'a (add (num 3) (num 4)))
```

```
subst (e') = (add (subst (id 'a)) (subst ((add (id 'b) (sub (num 35) (num 57)))))
           = (add (add (num 3) (num 4)) ((add (subst (id 'b)) (subst (sub (num 35) (num 57)))))
           = (add (add (num 3) (num 4)) ((add (id 'b) (sub (subst (num 35)) (subst (num 57)))))
           = (add (add (num 3) (num 4)) ((add (id 'b) (sub (num 35) (num 57)))).
```

b) $e = \{ \text{with } \{y \{-30 \{-y z\}\} \{-30 \{+y z\}\} \}$
 $(\text{subst (parse e) 'y (id 'w)})$

Primero veamos como esta construida la representación abstracta de la expresión anterior, esta es



Luego, apliquemos la instrucción parse a la expresión e, esto es

```
(parse e) = (with (parse { y {- 30 {- y z } } } (parse {- 30 {+ y z } })))
           = (with (((parse y) (parse {- 30 {- y z } })))
               (sub (parse 30) (parse {+ y z })))
           = (with ((id 'y) (sub (parse 30) (parse {- y z }))))
               (sub (num 30) (add (parse y) (parse z)))
           = (with ((id 'y) (sub (num 30) (sub (parse y) (parse z)))))
           = (sub (num 30) (add (id 'y) (id 'z)))
           = (with ((id 'y) (sub (num 30) (sub (id 'y) (id 'z)))))
           = (sub (num 30) (add (id 'y) (id 'z)))
           = e'
```

Ahora, apliquemos la instrucción subst a e' y los valores indicados, esto es (se obvia la observación anterior)

```

(subst e) = (subst (with ((id 'y) (sub (num 30) (sub (id 'y) (id 'z)))))
              (sub (num 30) (add (id 'y) (id 'z)))))
= (with (subst ((id 'y) (sub (num 30) (sub (id 'y) (id 'z)))))
        (subst (sub (num 30) (add (id 'y) (id 'z)))))
= (with ((subst (id 'y)) (subst (sub (num 30) (sub (id 'y) (id 'z)))))
        (sub (subst (num 30)) (subst (add (id 'y) (id 'z)))))
= (with ((id 'y) (sub (subst (num 30)) (subst (sub (id 'y) (id 'z)))))
        (sub (num 30) (add (subst (id 'y)) (subst (id 'z)))))
= (with ((id 'w) (sub (num 30) (sub (subst (id 'y)) (subst (id 'z)))))
        (sub (num 30) (add (id 'w) (id 'z)))))
= (with ((id 'w) (sub (num 30) (sub (id 'w) (id 'z)))))
        (sub (num 30) (add (id 'w) (id 'z)))).

```

c) $e = \{ \text{with}\{y \{-30 \{-y \ z\}\} \}$
 $\{-30 \{+y \ z\}\}$
 $(\text{subst} (\text{parse } e) \text{'z } (\text{id 'v}))$

Observemos que la representación abstracta para este inciso es la misma que para el inciso anterior, pues la expresión `with`, en principio, es la misma. Así, omitimos este paso.

Aplicar la instrucción `parse` para este inciso nos devuelve lo mismo que en el inciso anterior, así omitimos este paso y tomaremos el valor de `e'` encontrado en el inciso previo.

Ahora, apliquemos la instrucción `subst` a `e'` y los valores indicados, esto es (se obvia la observación hecha en este ejercicio)

```

(subst e) = (subst (with ((id 'y) (sub (num 30) (sub (id 'y) (id 'z)))))
              (sub (num 30) (add (id 'y) (id 'z)))))
= (with (subst ((id 'y) (sub (num 30) (sub (id 'y) (id 'z)))))
        (subst (sub (num 30) (add (id 'y) (id 'z)))))
= (with ((subst (id 'y)) (subst (sub (num 30) (sub (id 'y) (id 'z)))))
        (sub (subst (num 30)) (subst (add (id 'y) (id 'z)))))
= (with ((id 'y) (sub (subst (num 30)) (subst (sub (id 'y) (id 'z)))))
        (sub (num 30) (add (subst (id 'y)) (subst (id 'z)))))
= (with ((id 'y) (sub (num 30) (sub (subst (id 'y)) (subst (id 'z)))))
        (sub (num 30) (add (id 'y) (id 'v)))))
= (with ((id 'y) (sub (num 30) (sub (id 'y) (id 'v)))))
        (sub (num 30) (add (id 'y) (id 'v)))).

```