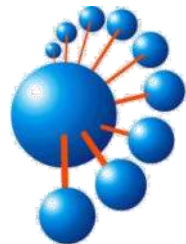


# Inteligencia Artificial

## Métodos de búsqueda informada

Profesor: Julio Godoy  
Ayudante: Felipe Cerda  
DIICC





# BÚSQUEDA INFORMADA (HEURÍSTICA)

- Se basa en una función de evaluación  $f$  que da una indicación acerca del mejor nodo a ser expandido
  - familia de métodos de búsqueda con diversas  $f$
  - estimación de la distancia al objetivo : *heurística*  $h(n)$
  - $h(n)$  = costo estimado del 'mejor' camino desde el nodo actual hasta la meta
- El nodo con el valor más bajo se expande primero

función BEST-FIRST-SEARCH(*problem*, EVAL) returns *solution o fracaso*

*Fila* := función que entrega los nodos ordenados por EVAL

return GENERAL-SEARCH(*problem*, *Fila*)

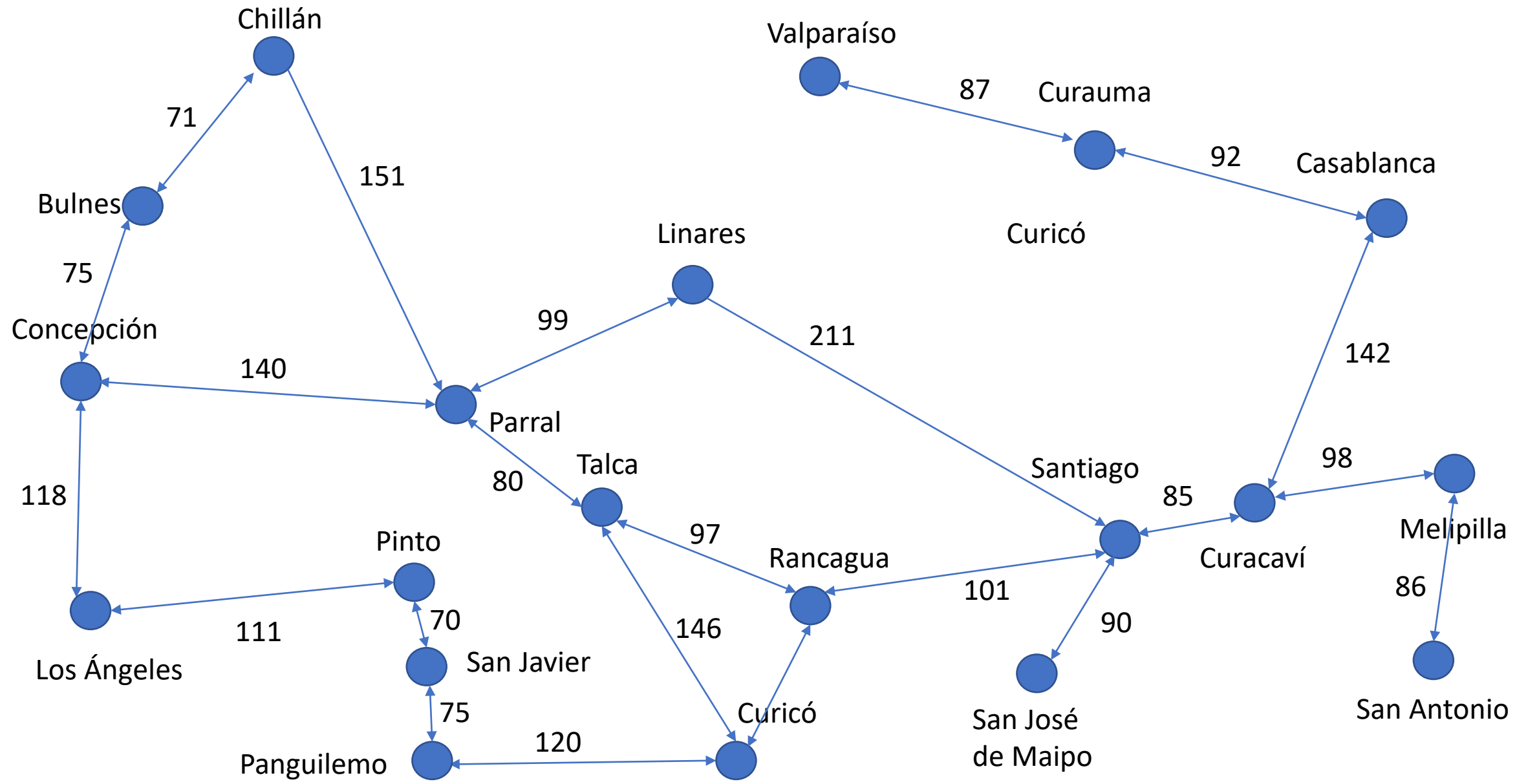


# BÚSQUEDA GREEDY BEST FIRST SEARCH

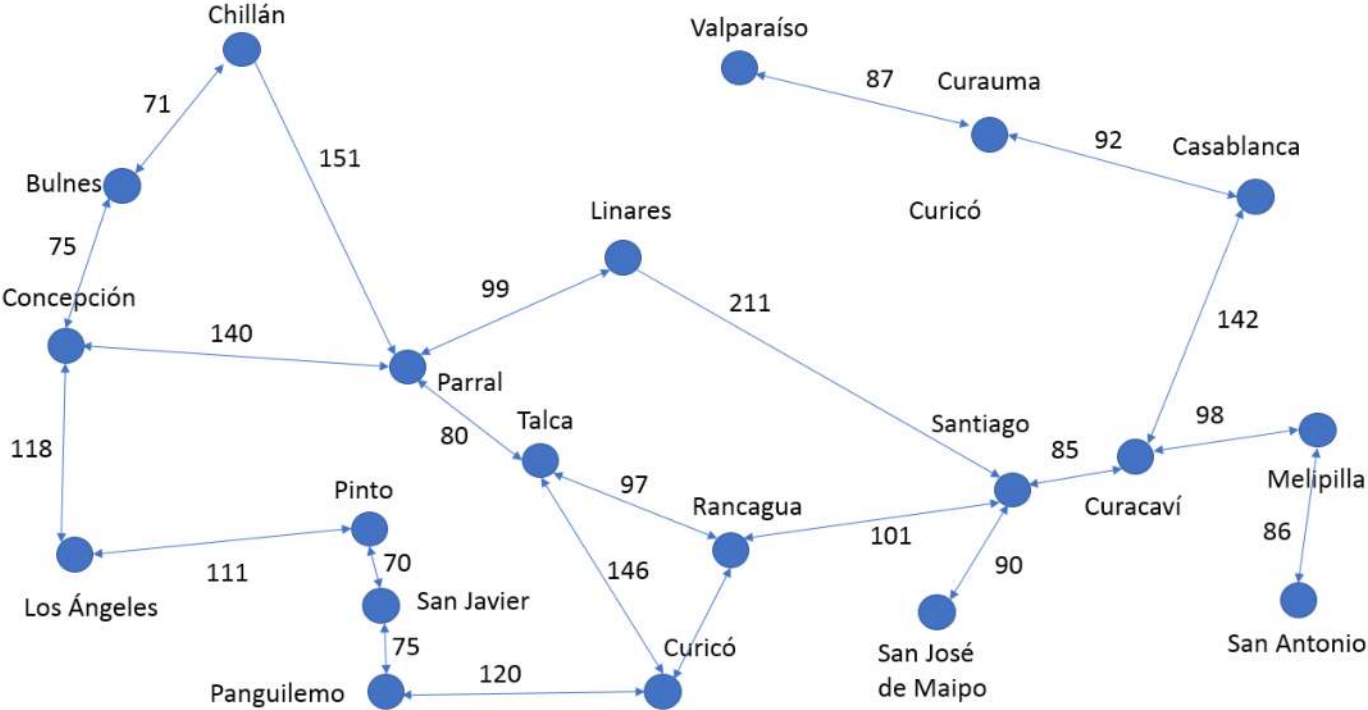
- Minimiza el costo estimado a un objetivo
  - expande el nodo que parece más cercano a la meta
  - utiliza una función heurística
    - $h(n)$  = costo estimado desde el nodo actual hasta la meta
    - las funciones heurísticas son propias de un problema específico
    - a menudo la distancia en línea recta para encontrar rutas y problemas similares
- a menudo mejor que búsqueda en profundidad, aunque las peores complejidades en tiempo son iguales o peores en espacio

función GREEDY-SEARCH(*problema*) returns *solución o fracaso*

return BEST-FIRST-SEARCH(*problema*, *h*)



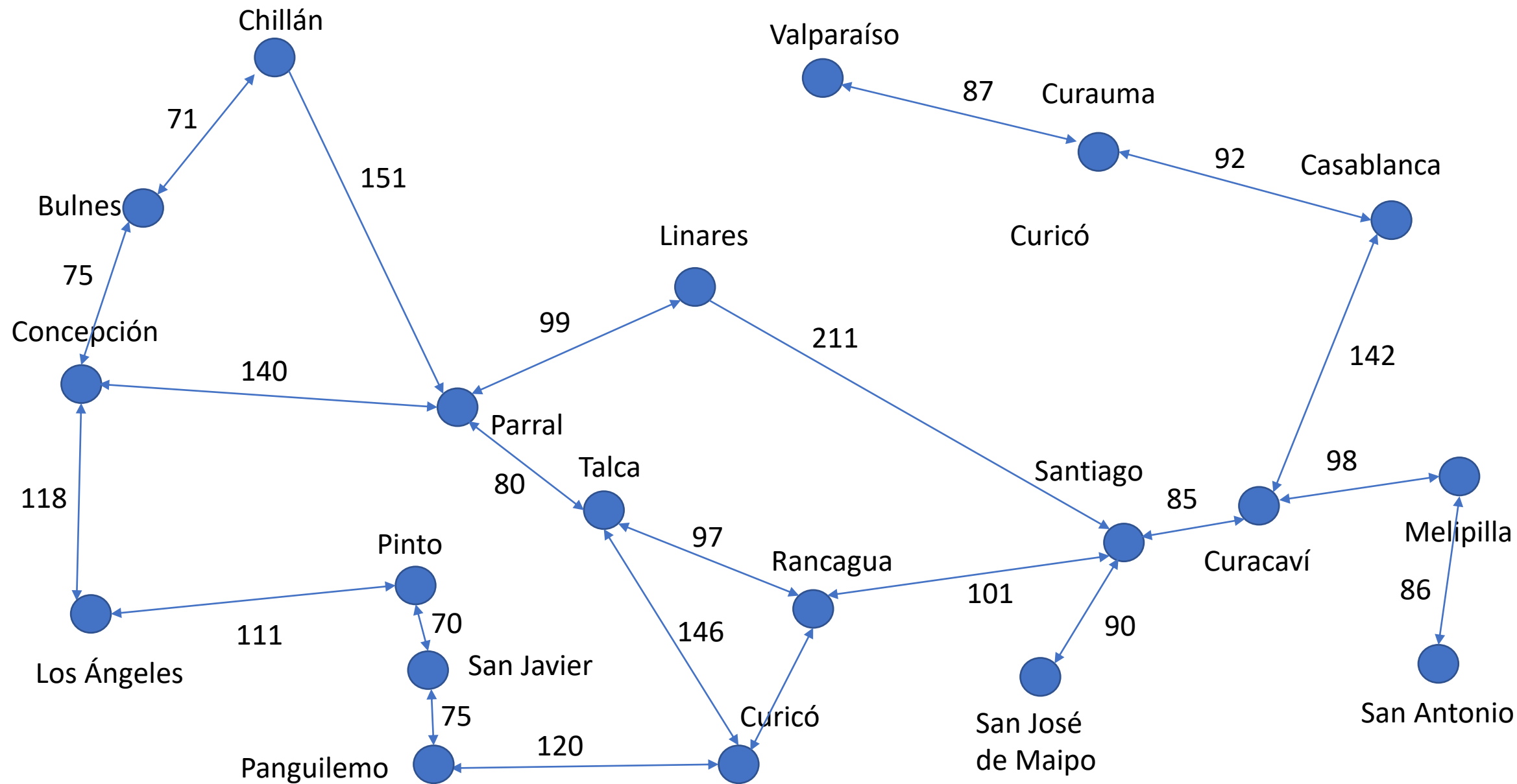
estado	$h(\text{estado})$
Concepción	450
Los Ángeles	520
Valparaíso	120
Parral	350
Bulnes	400
Linares	300
Chillán	380



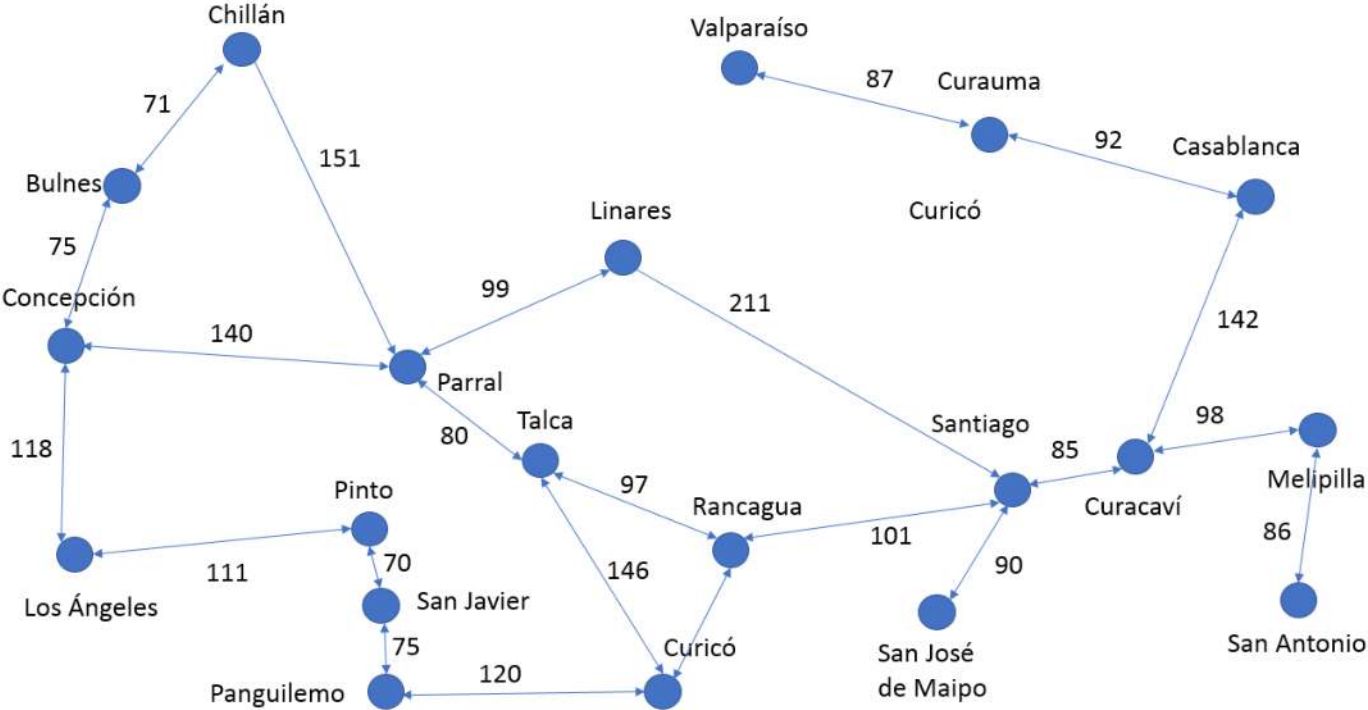


# BÚSQUEDA A\*

- Búsqueda A\* (A “estrella”):
  - Decide a qué estado moverse en base a una función de evaluación  $f$
  - $f(\text{estado}) = \text{costo para llegar al estado} + \text{costo aproximado del camino más corto hasta el objetivo (heurística)}$
  - $f(\text{estado}) = g(\text{estado}) + h(\text{estado})$
- El nodo que representa al estado con el valor más bajo se expande primero



estado	$h(\text{estado})$
Concepción	450
Los Ángeles	520
Valparaíso	120
Parral	350
Bulnes	400
Linares	300
....	.....







Concepción



Concepción

$f(\text{Concepción}) = \text{costo en llegar a Concepción} + \text{valor de heurística}$



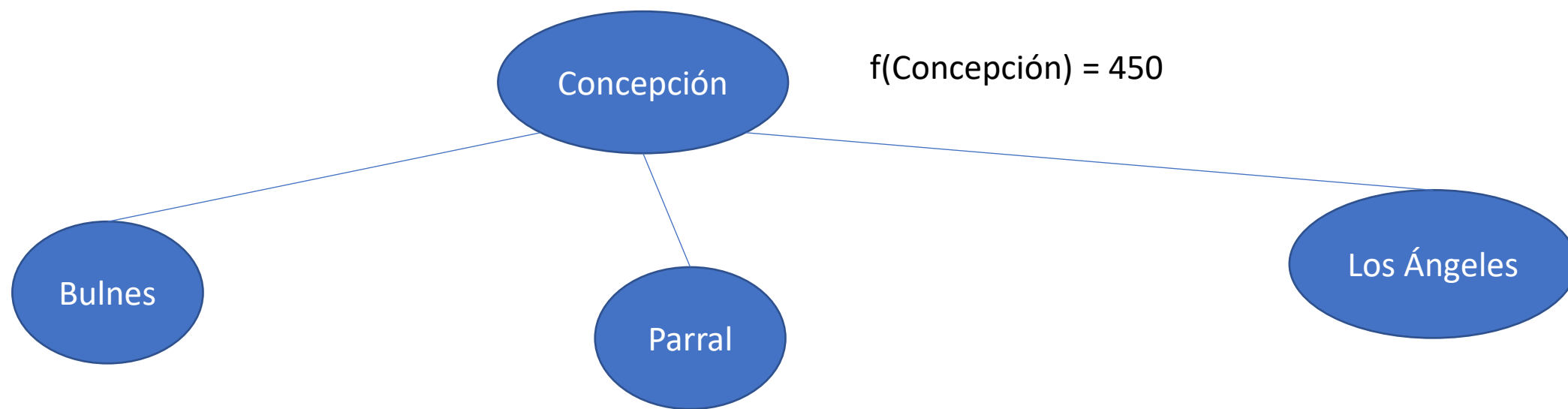
Concepción

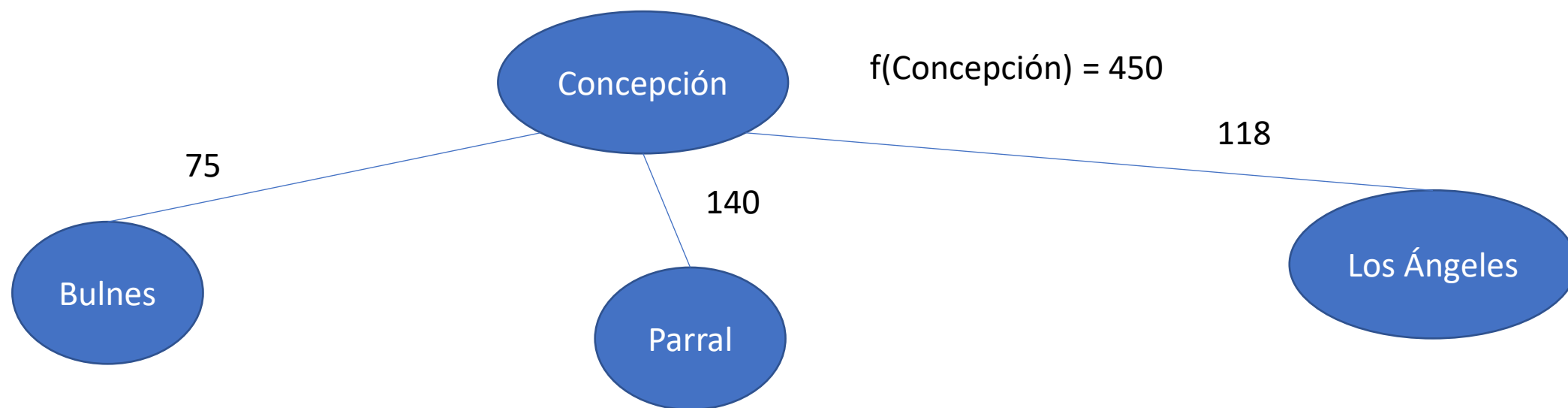
$f(\text{Concepción}) = 0 + \text{valor de heurística}$

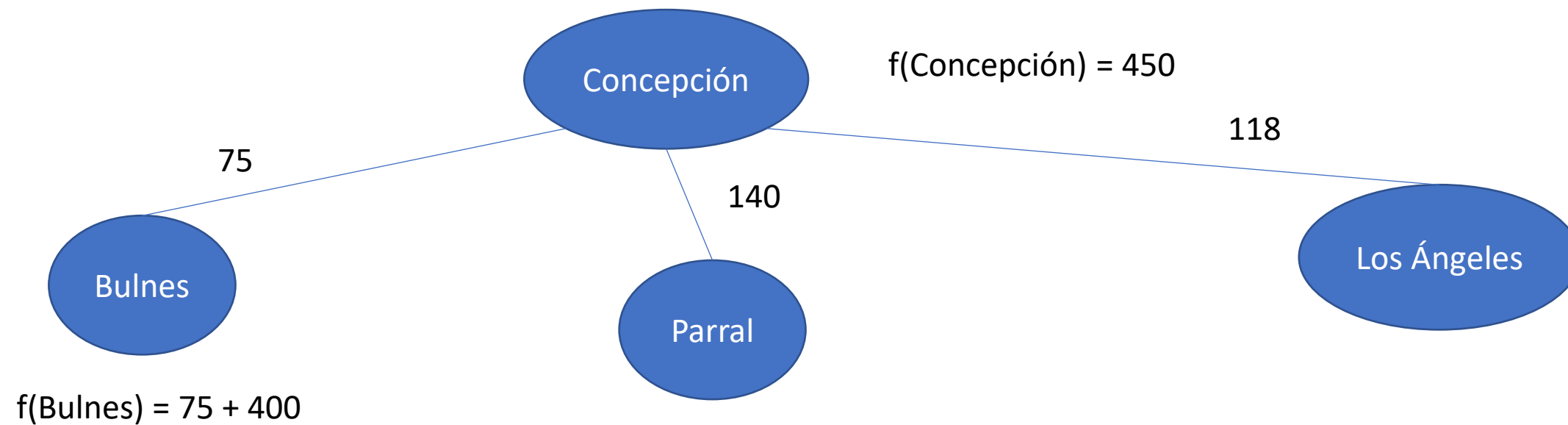


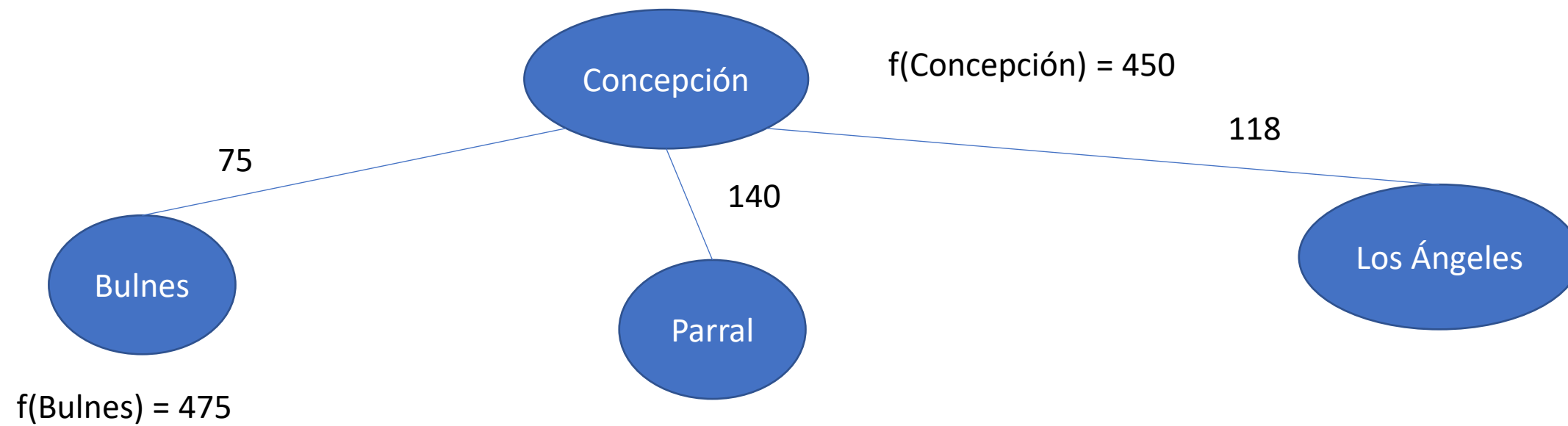
Concepción

$$f(\text{Concepción}) = 0 + 450 = 450$$

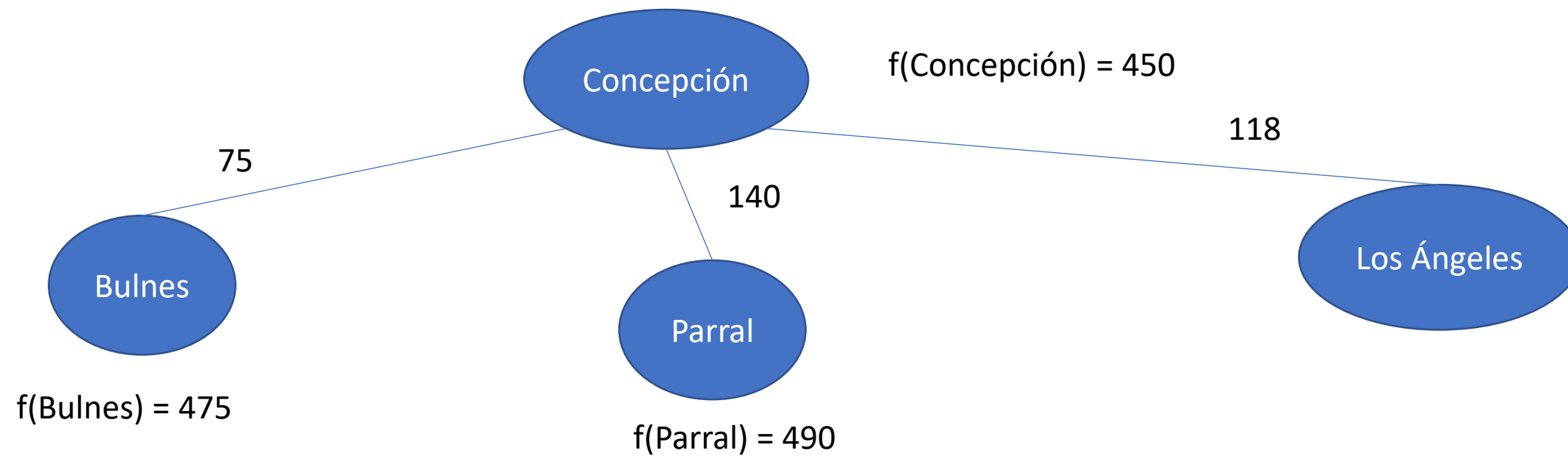


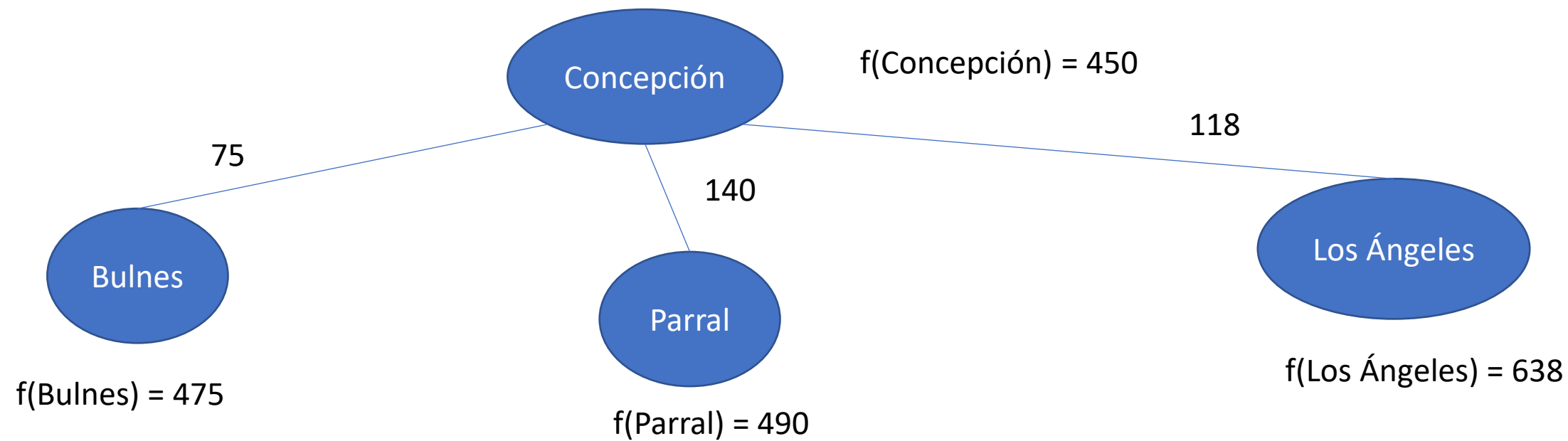


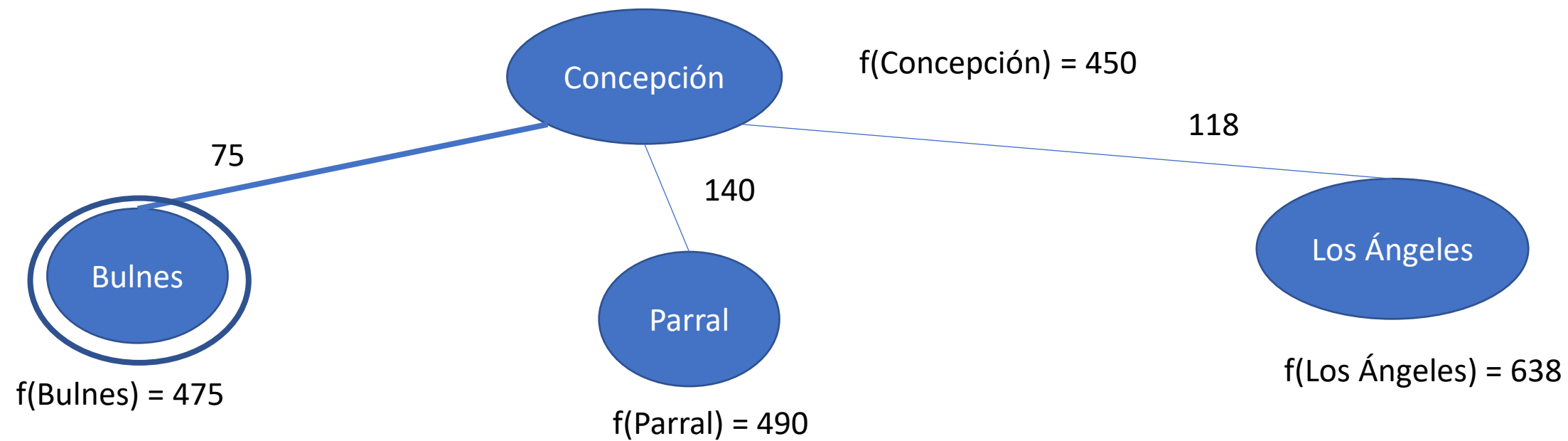


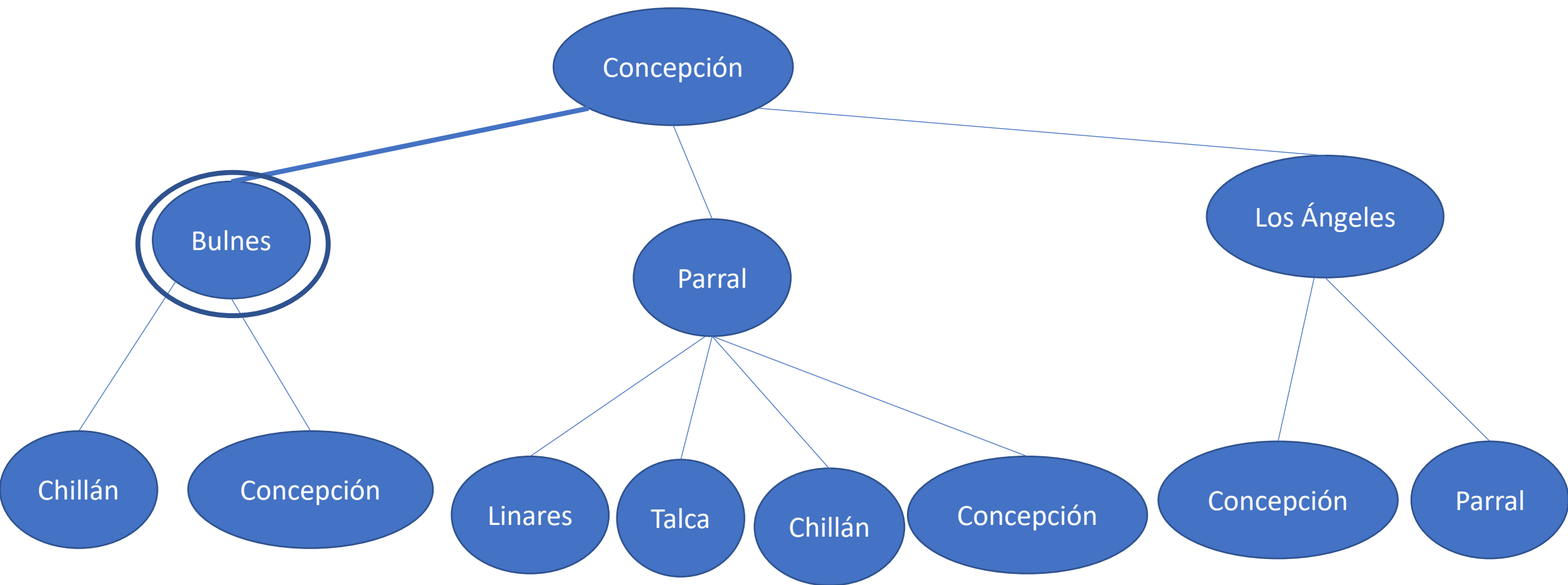


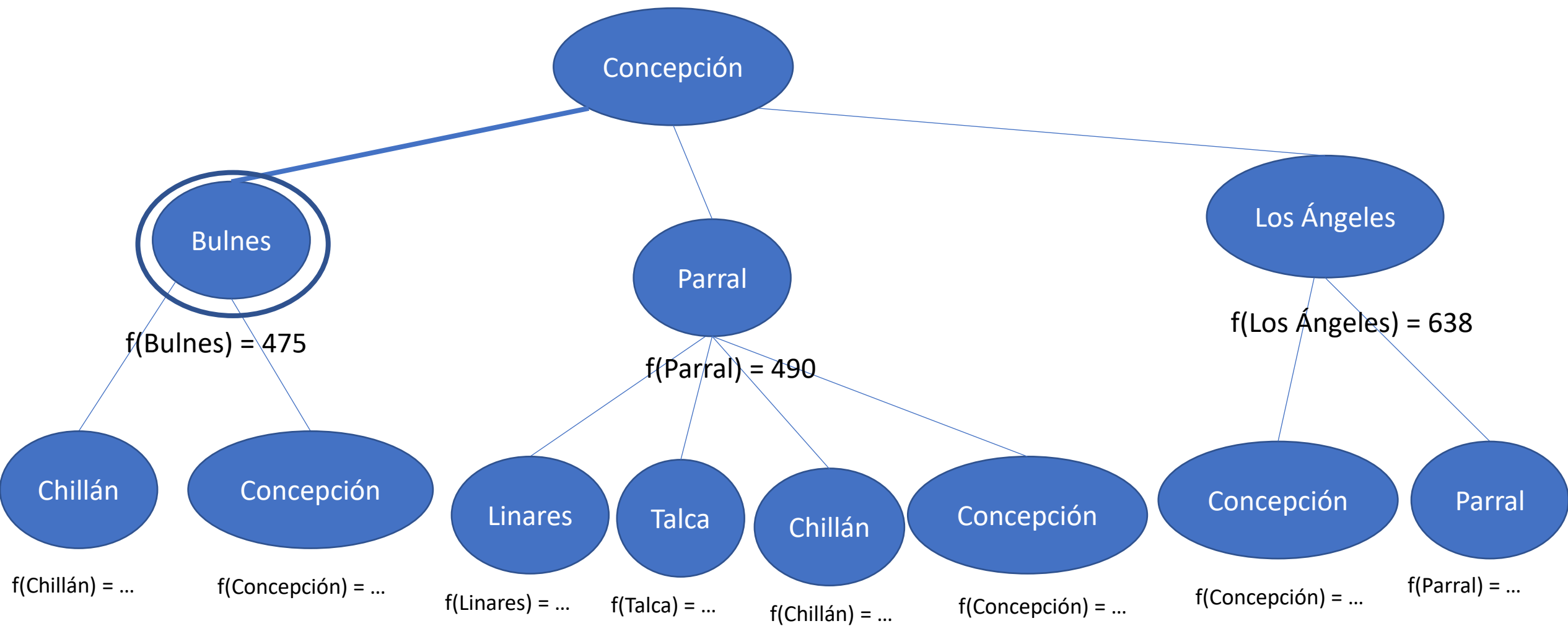








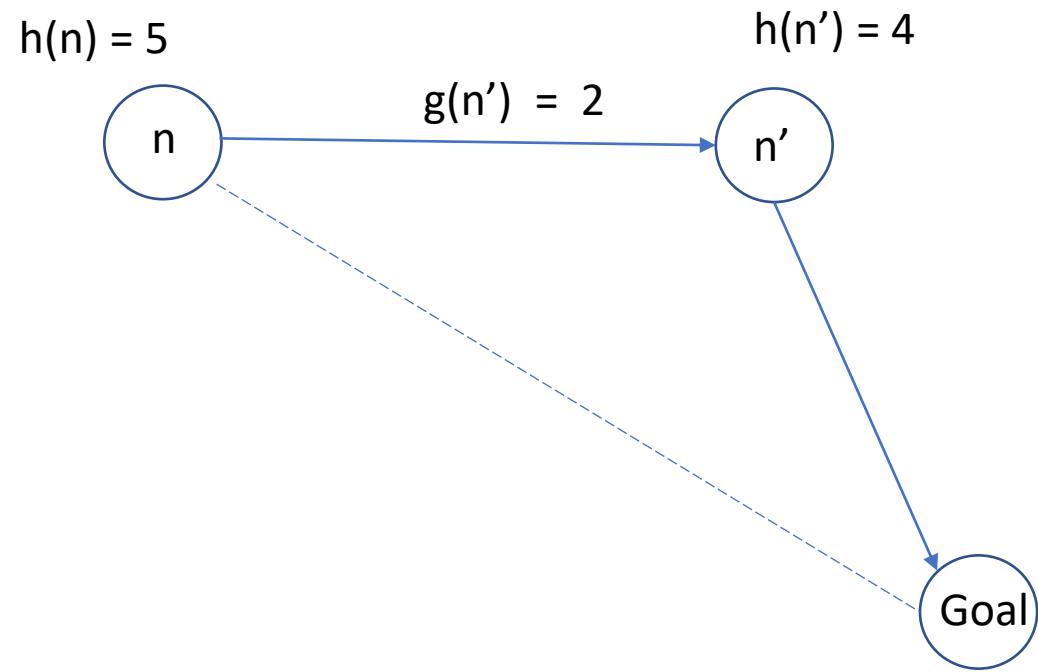




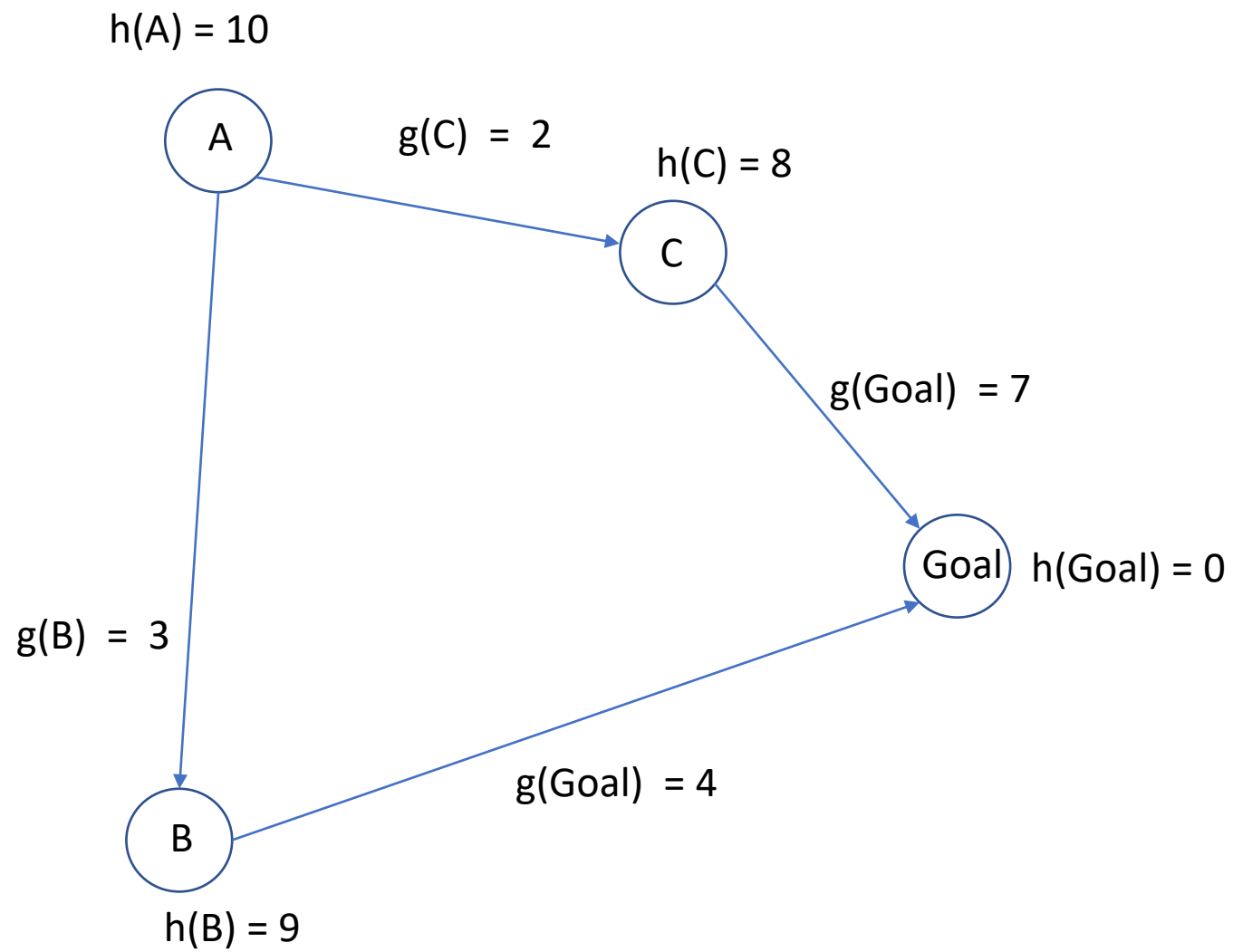


# PROPIEDADES DE LAS HEURÍSTICAS

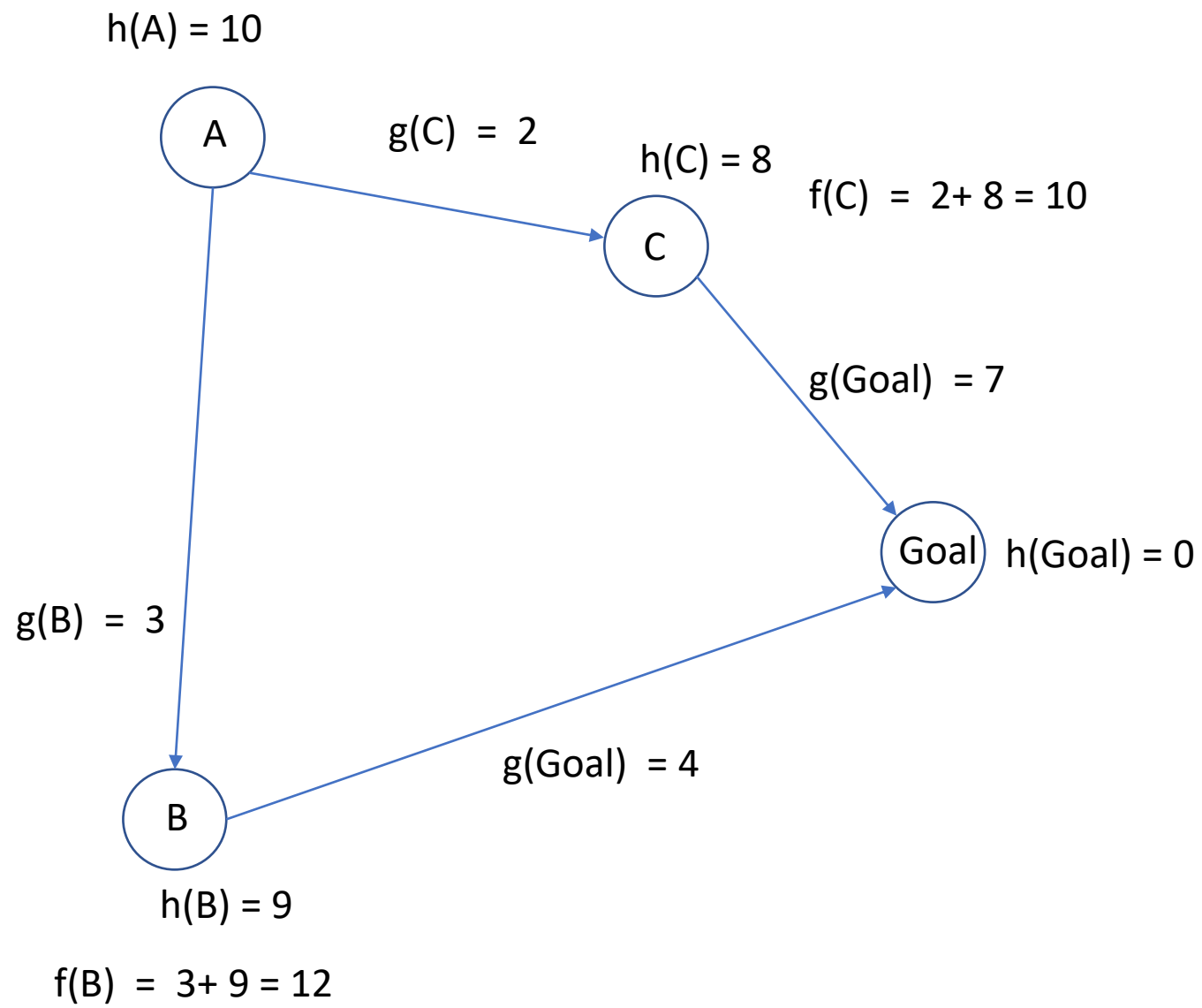
- Por definición, la heurística es una estimación del menor costo al objetivo, a partir de algún nodo  $n$ 
  - Esta estimación puede ser buena (menor o igual al costo real) o mala (mayor al costo real)
- **Admisibilidad:** para cada nodo  $n$ , heurística sub-estima (o es idéntica) al costo real
  - Condición necesaria para optimalidad de  $A^*$  en un árbol de búsqueda
- **Consistencia:** Para cada nodo  $n'$  sucesor de  $n$  (generado por alguna acción  $a$ ), el costo estimado de llegar al objetivo a partir de  $n$  no es mayor que el costo de llegar a  $n'$  más el costo de llegar al objetivo desde  $n'$ :
  - $h(n) \leq c(n, a, n') + h(n')$

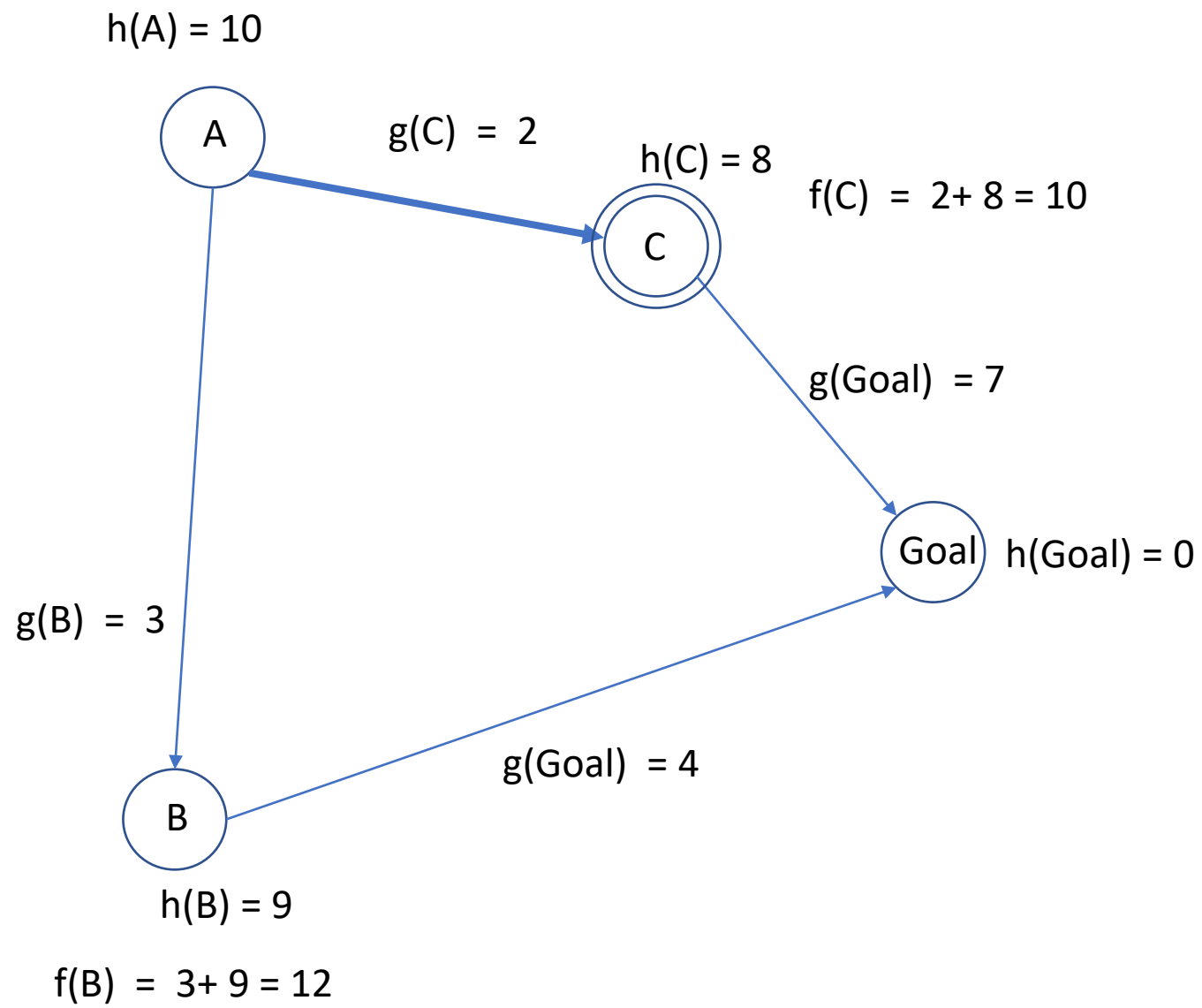


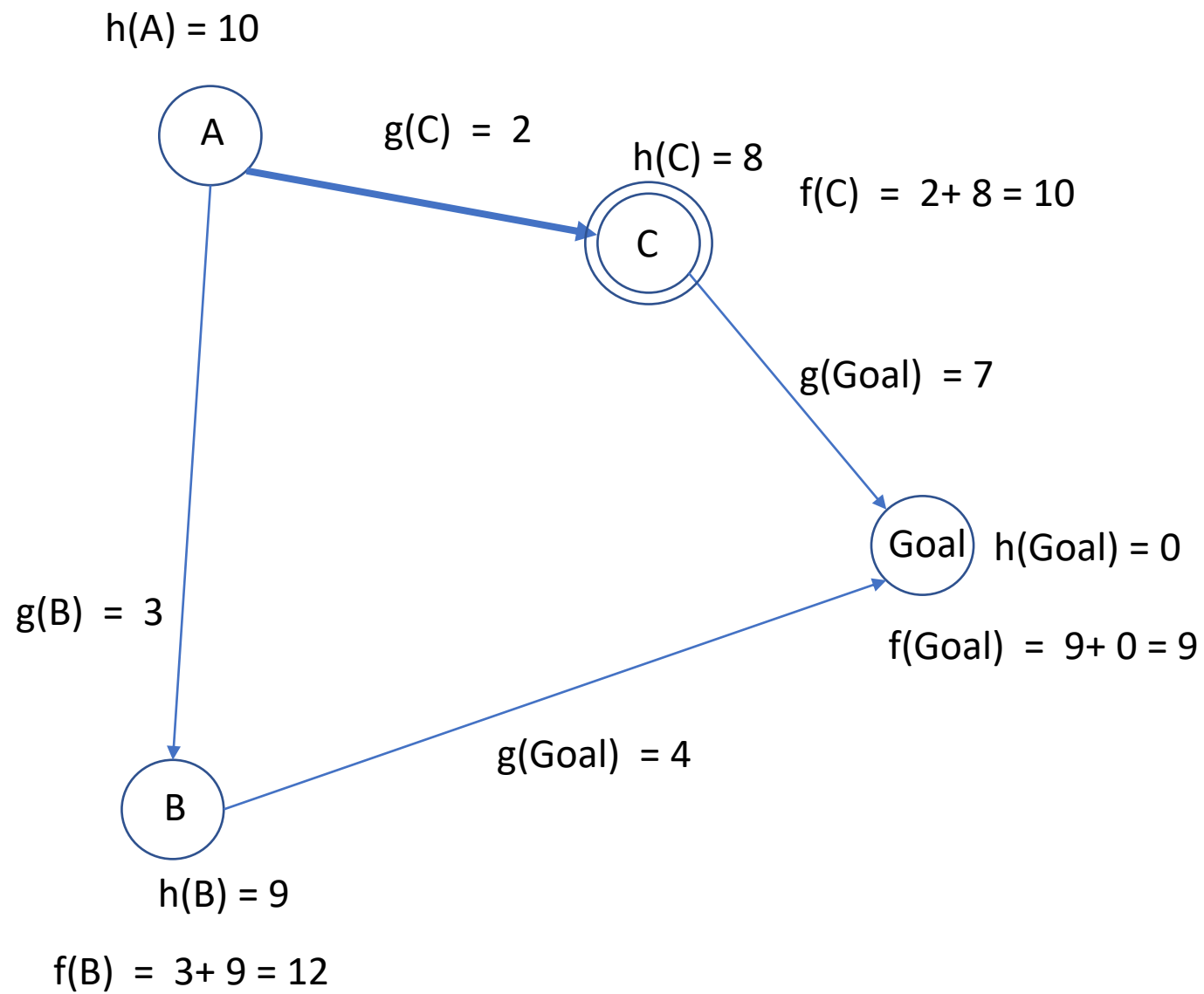
$$h(n) \leq c(n, a, n') + h(n')$$
$$5 \leq 2 + 4$$

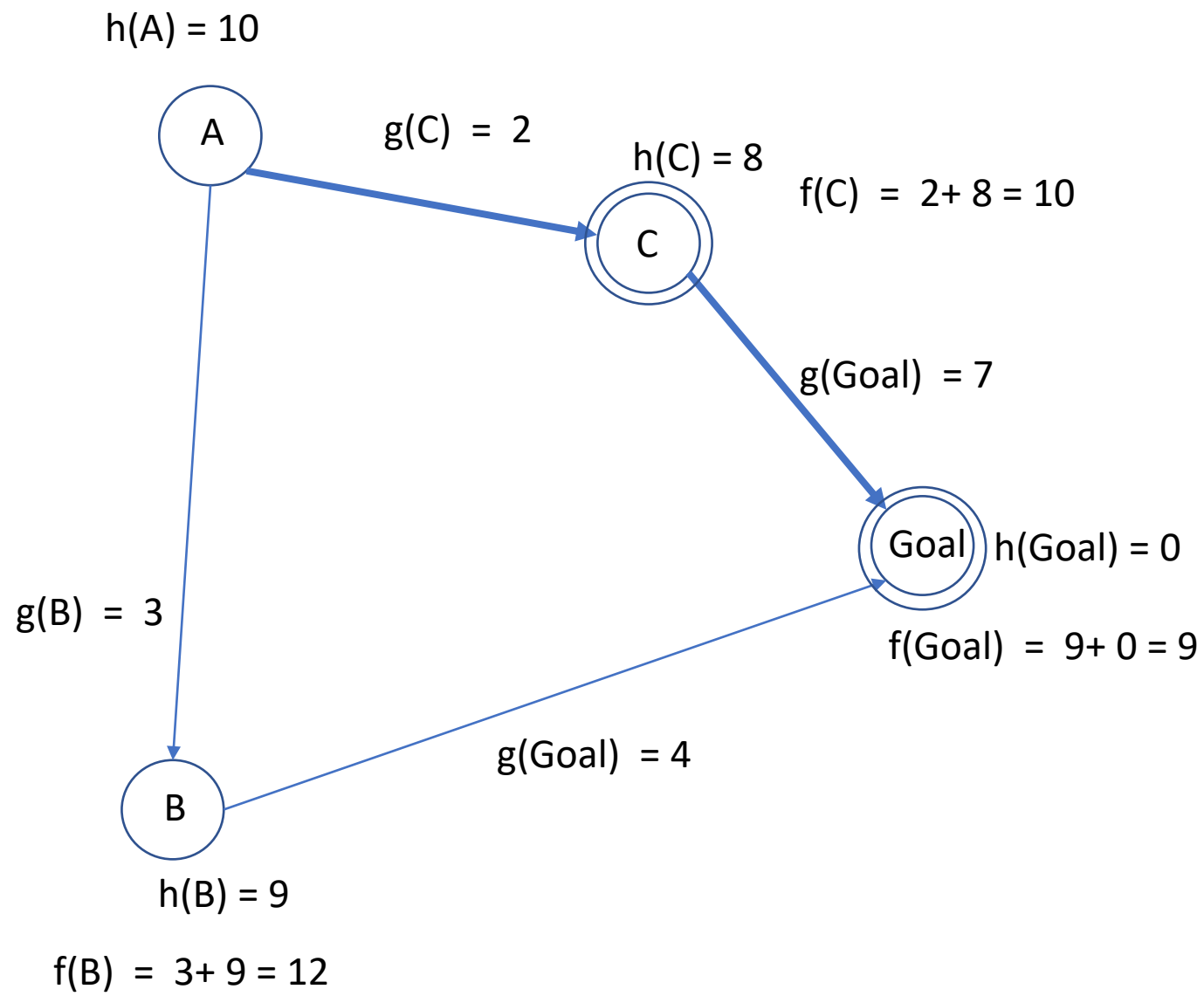












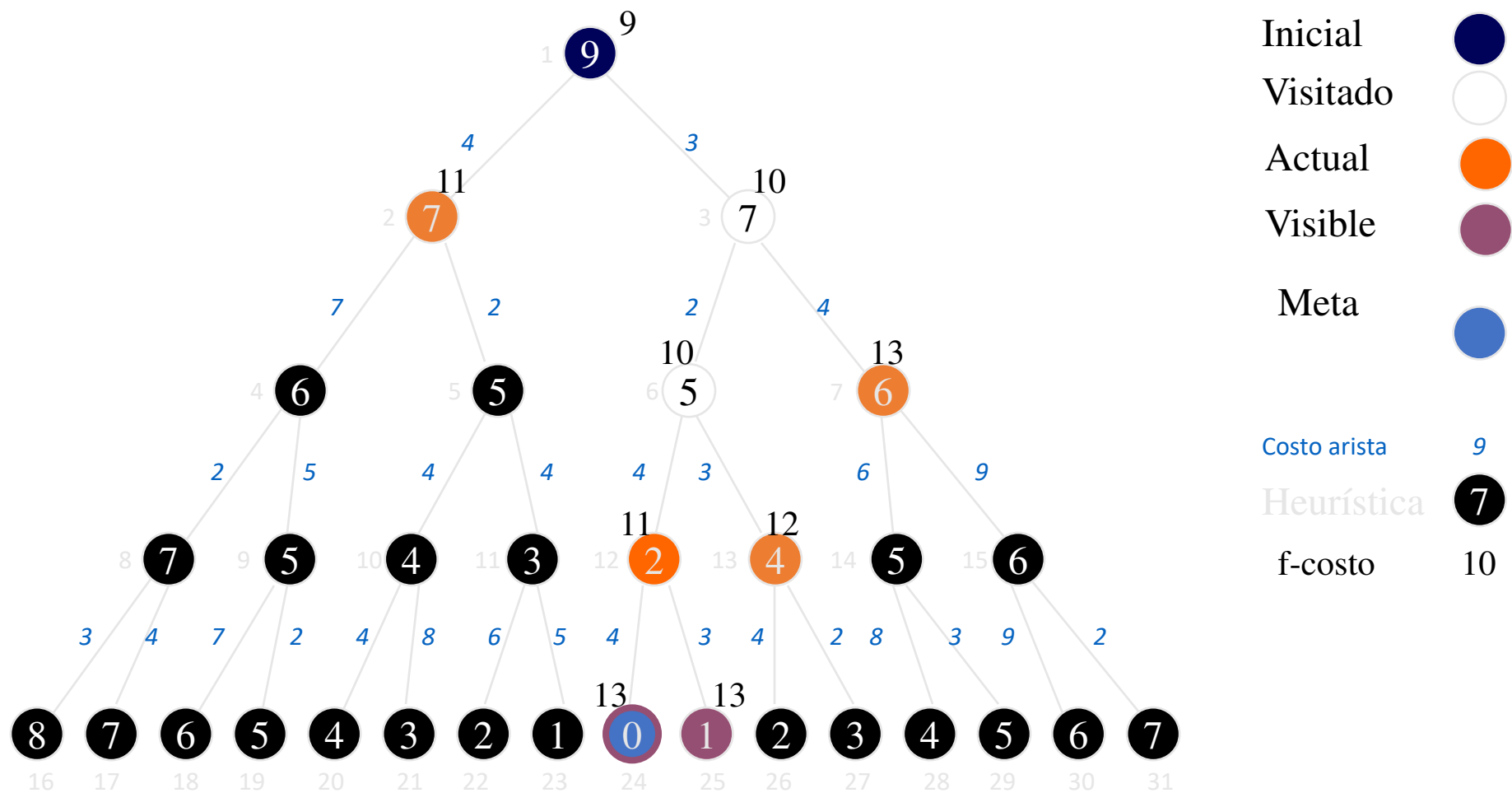


# PROPIEDADES DE $A^*$

- El valor de  $f$  nunca disminuye a lo largo del camino desde el nodo inicial
  - también se conoce como ***monotonidad*** de la función
  - casi todas las heurísticas admisibles muestran monotonidad
    - las que no, pueden ser modificadas sin mucho esfuerzo
- esta propiedad puede ser usada para generar contornos
  - regiones donde el f-costo está bajo un cierto umbral
  - con costo uniforme ( $h = 0$ ), los contornos son circulares
  - a mejor heurística  $h$ , más preciso es el contorno en torno al camino óptimo



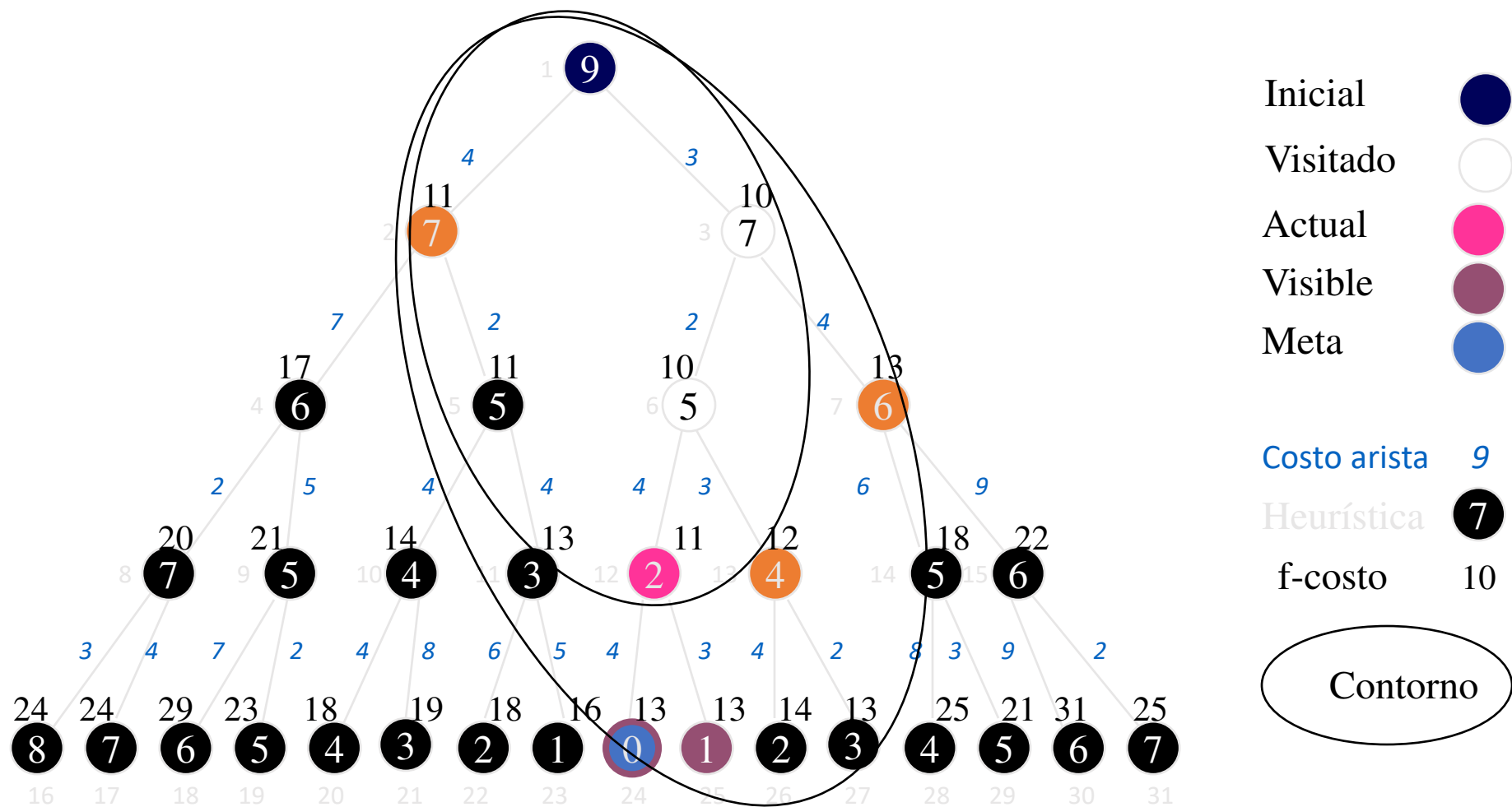
# INSTANTÁNEA DE A\*



nodos: [2(4+7), 7(3+4+6), 13(3+2+3+4)]



# INSTANTÁNEA DE A\* CON CONTORNOS





# OPTIMALIDAD DE $A^*$

- $A^*$  encontrará la solución optimal
  - la primera solución encontrada es la optimal
- $A^*$  es optimalmente eficiente
  - no hay otro algoritmo que garantice la expansión de menos nodos que  $A^*$





# COMPLEJIDAD DE $A^*$

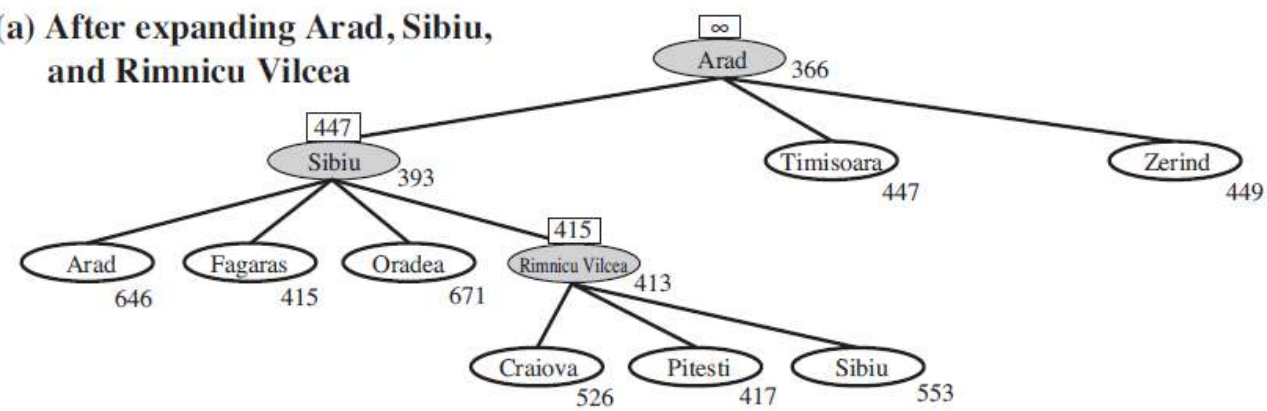
- El número de nodos dentro del espacio de búsqueda todavía es exponencial
  - con respecto a la longitud de la solución
  - mejor que otros algoritmos, pero todavía problemático
- con frecuencia, la complejidad espacial es más severa que la complejidad temporal
  - $A^*$  guarda todos los nodos generados en memoria



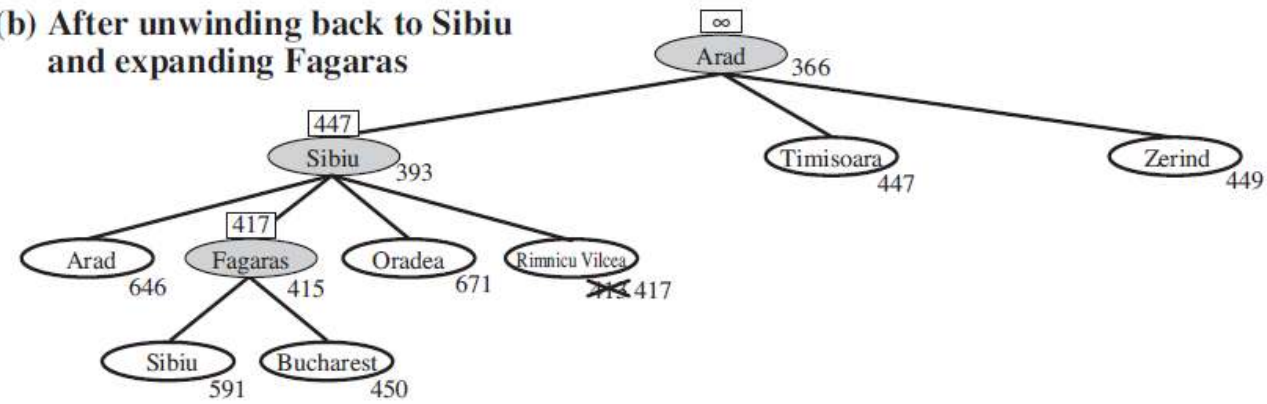
# BÚSQUEDA CON MEMORIA ACOTADA

- Algoritmos de búsqueda que tratan de ahorrar memoria
- La mayoría son variaciones de  $A^*$ 
  - profundización iterativa de  $A^*$  (IDA\*)
  - Búsqueda best first recursiva (RBFS)
  - memoria acotada simplificada  $A^*$  (SMA\*)

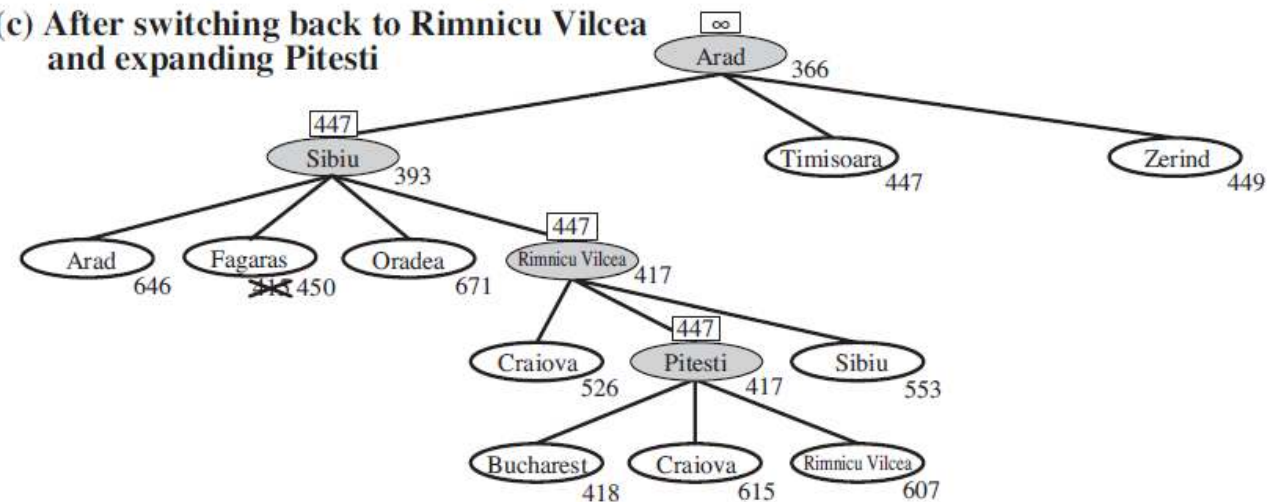
(a) After expanding Arad, Sibiu, and Rimnicu Vilcea



(b) After unwinding back to Sibiu and expanding Fagaras



(c) After switching back to Rimnicu Vilcea and expanding Pitesti





# HEURÍSTICAS DE BÚSQUEDA

- Para muchas tareas, una buena heurística es la clave para encontrar una solución
  - Reduce el tamaño del espacio de búsqueda
  - Orienta la búsqueda hacia la meta
- Relajación de problemas
  - Menos restricciones a los operadores
  - Su solución exacta puede ser una buena heurística para el problema original



---

$A^*$  and Heuristics  
in 2 minutes

---



# HEURÍSTICA PUZZLE-8

- **Nivel de dificultad**
  - alrededor de 20 pasos para una solución típica
  - factor de ramificación aproximadamente 3
  - búsqueda exhaustiva lleva a  $3^{20} = 3.5 * 10^9$
  - $9! = 362,880$  diferentes configuraciones para las 9 piezas
- **candidatas a función heurística**
  - número de piezas fuera de lugar
  - suma de las distancias de las piezas a sus posiciones correctas
- **generación de heurísticas**
  - posible a partir de especificaciones formales

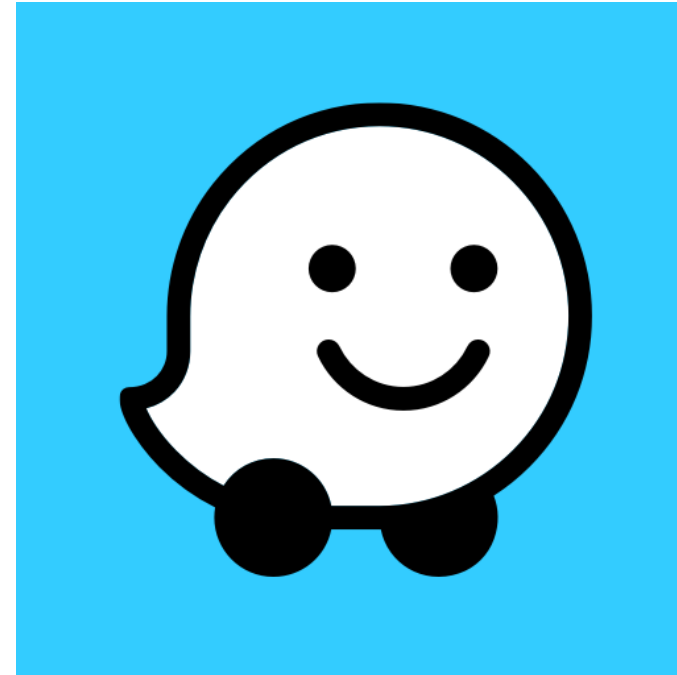


# APLICACIONES DE BÚSQUEDA INFORMADA

- Asistentes de navegación



Google Maps





# APLICACIONES DE BÚSQUEDA INFORMADA

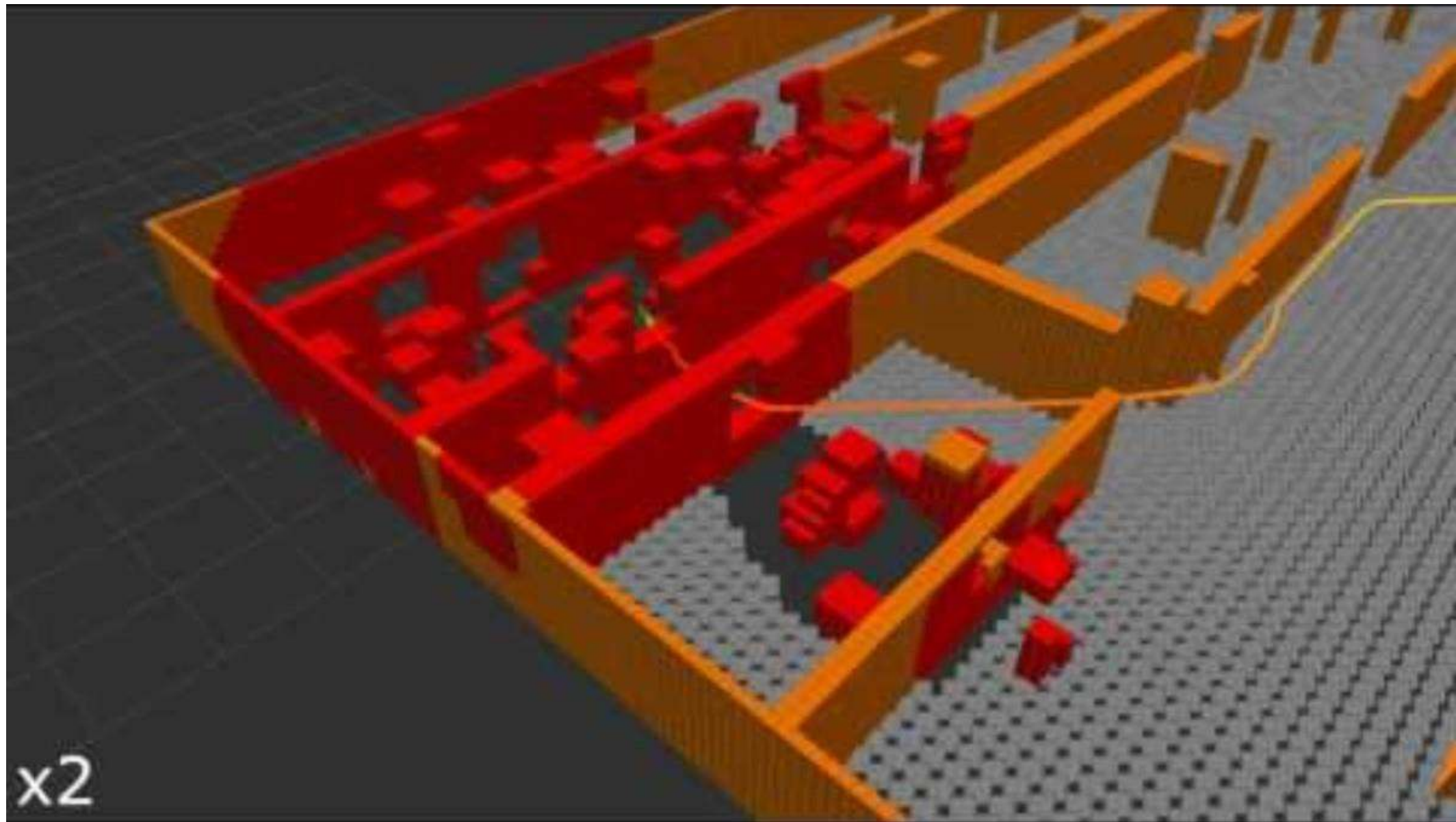
motion planning for autonomous vehicles in 4D ( $\langle x, y, \text{orientation}, \text{velocity} \rangle$ )  
running Anytime Incremental A\* (Anytime D\*) on multi-resolution lattice  
*[Likhachev & Ferguson, IJRR'09]*



*part of efforts by Tartanracing team from CMU for the Urban Challenge 2007 race*

Profesor Julio Godoy







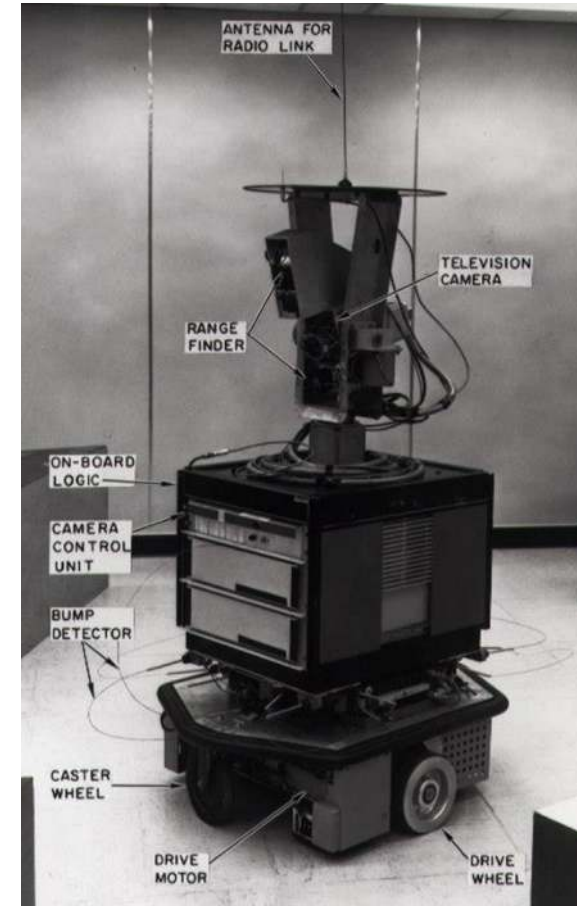
¿Weighted A\*?





# APLICACIONES DE BÚSQUEDA INFORMADA

- Shakey:
  - Primer robot móvil de propósito general
  - Utilizaba, entre otros programas, métodos de búsqueda para encontrar rutas de navegación

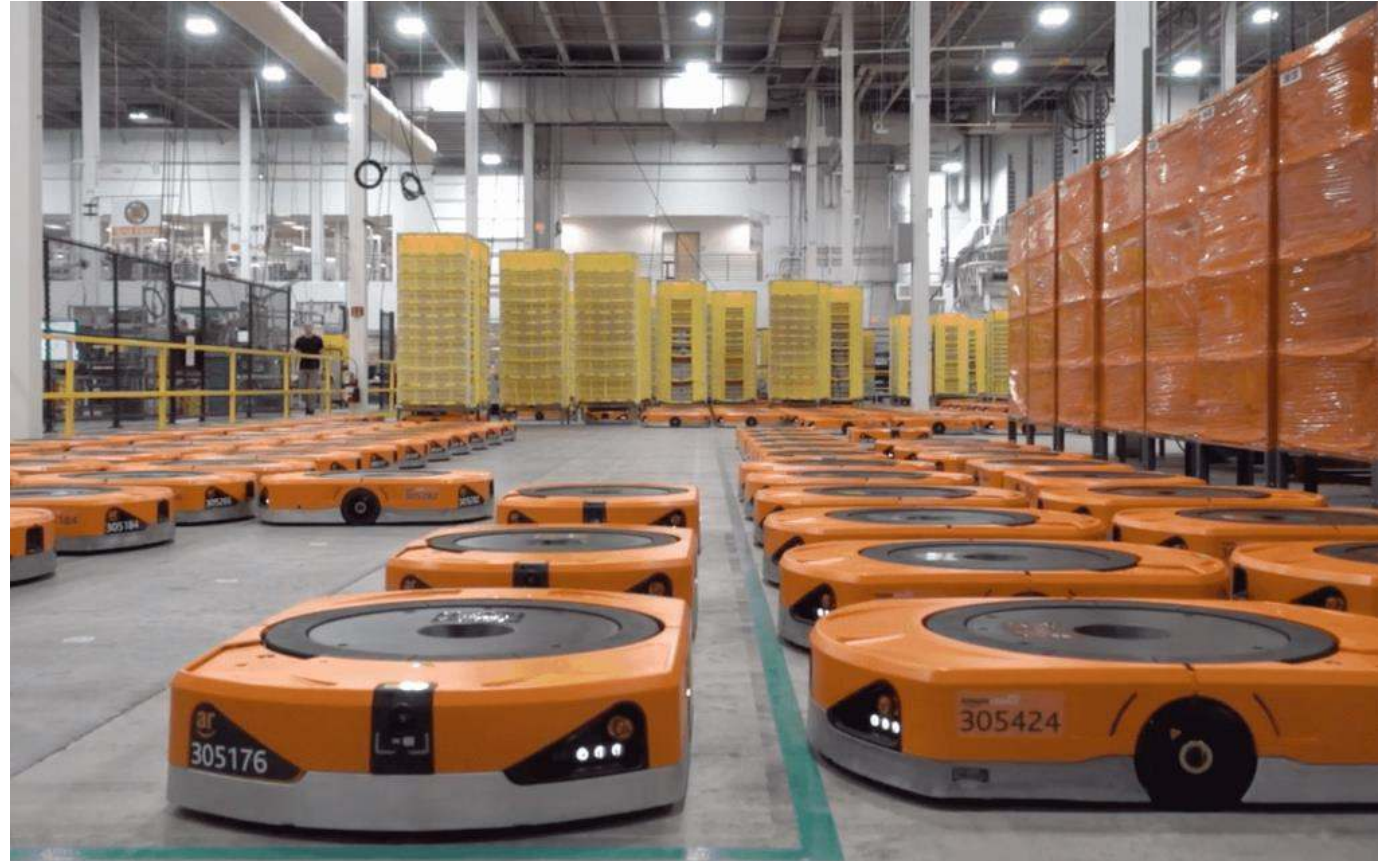






# APLICACIONES DE BÚSQUEDA INFORMADA

- Amazon robotics:





# AMAZON ROBOTICS

