

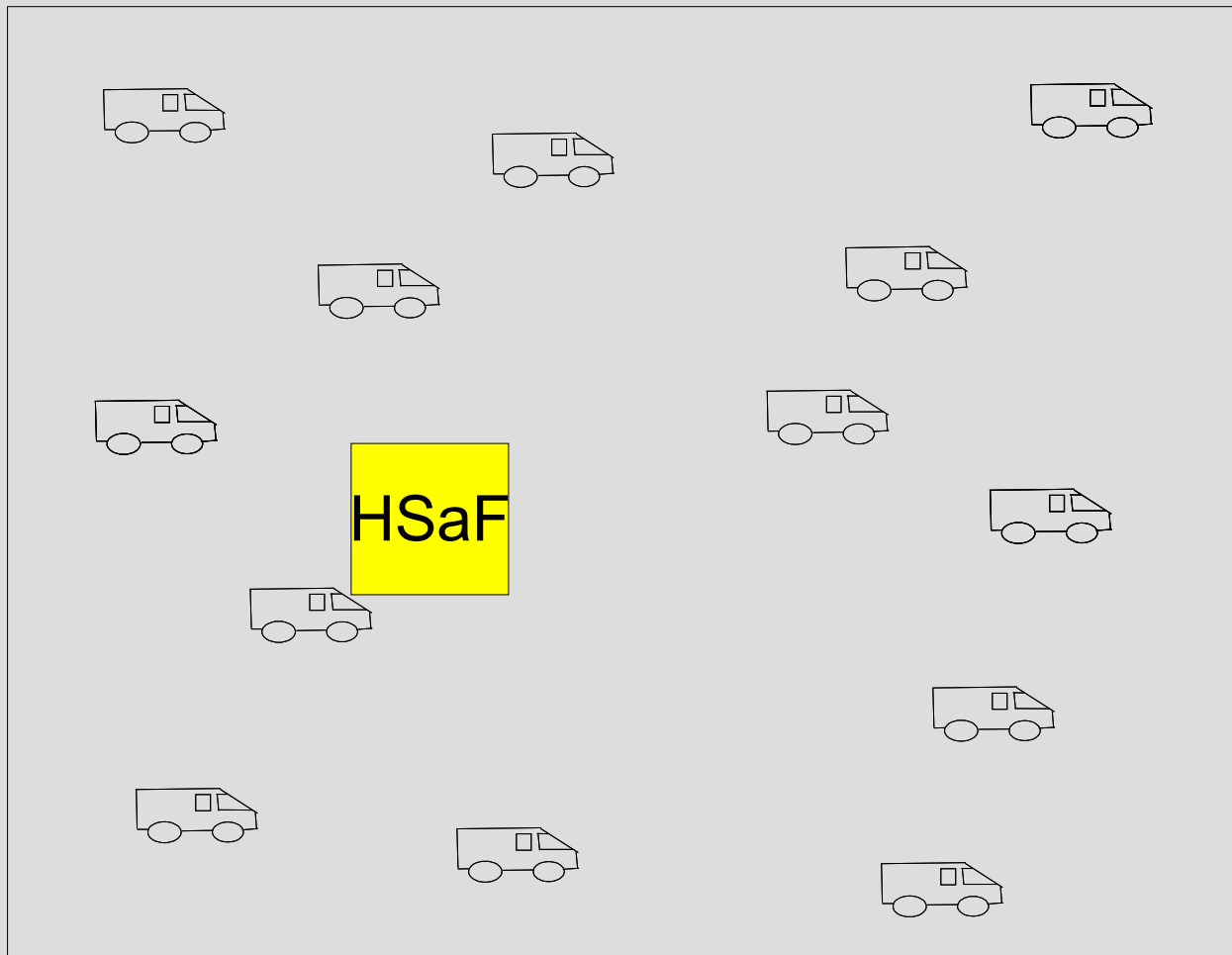
Verteilung und Integration von Informationen im Verkehrsbereich

Mobile Objekte – Indexstrukturen

Ingo Beutler

07.06.2004

Wie können räumliche Daten effizient gespeichert werden?



Anfragen:

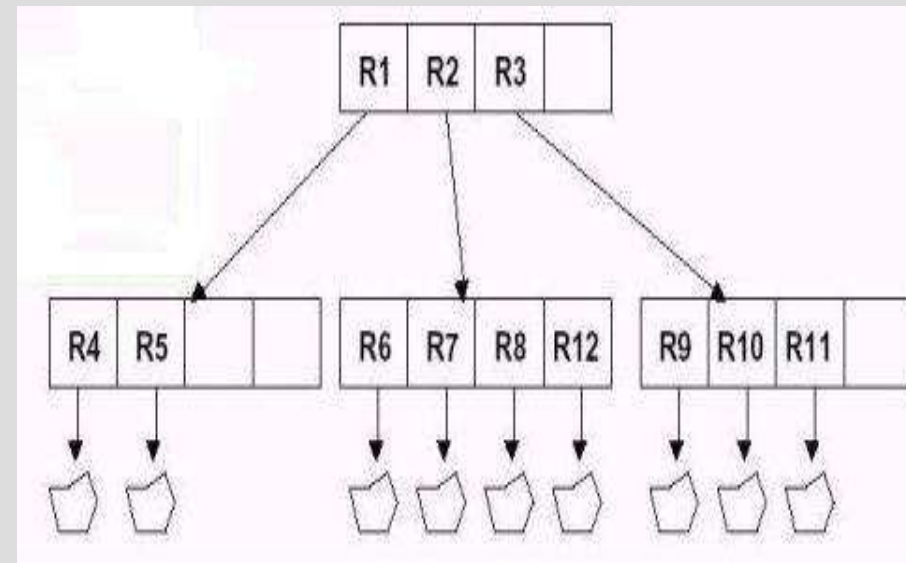
z.B. Welche Transporter befinden sich in der Nähe des HSaF?

Klassischer Ansatz

B-Baum

+ Indexstruktur zu Speicherung von großen Datenmengen

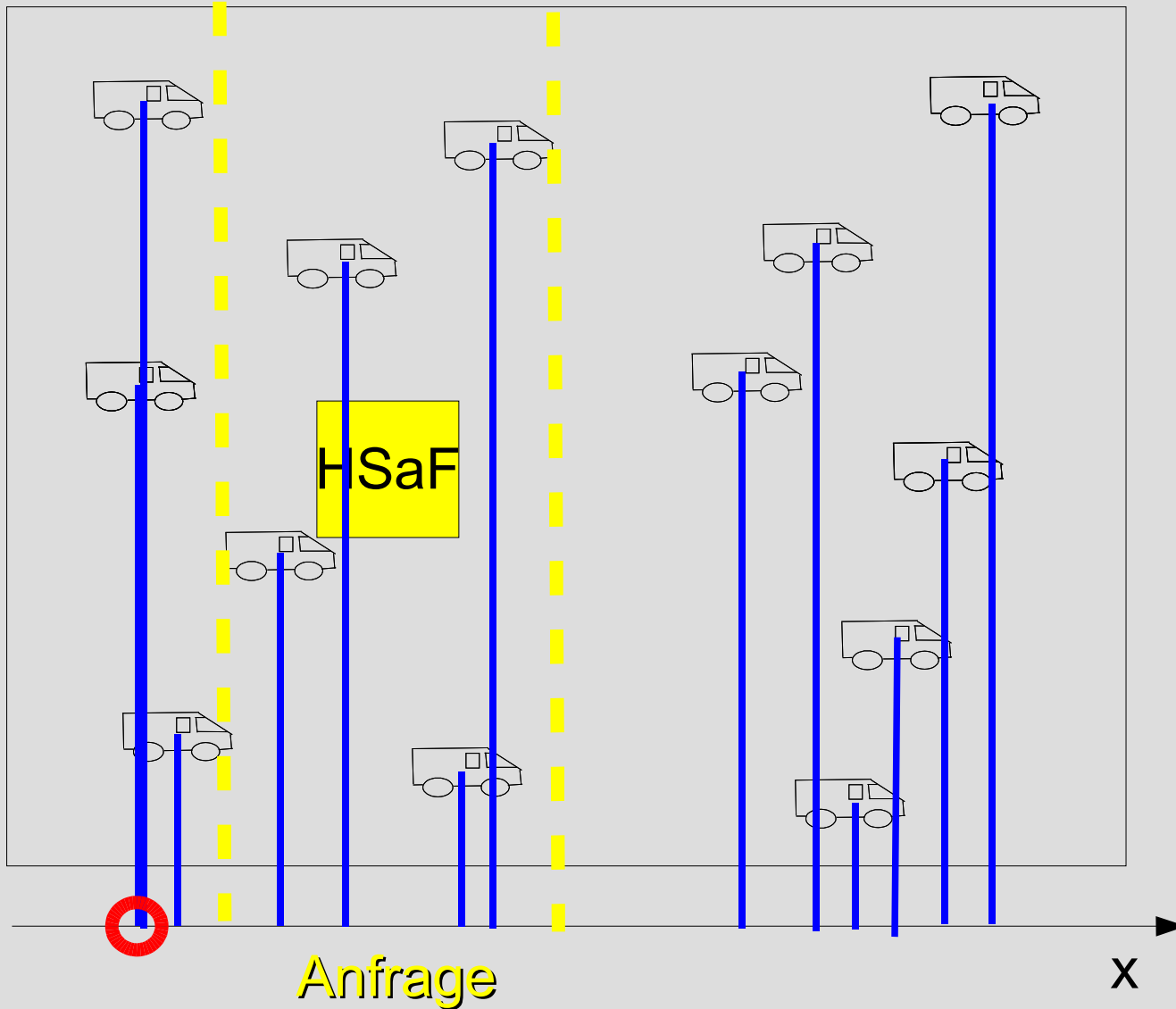
- ▶ Insert, Delete, Search
- ▶ vollständig ausgeglichen
- ▶ Knotengröße = Pagegröße
- ▶ Mindest-Füllgrad der Knoten



- Index eindimensional

→ Keine effiziente räumliche Suche möglich

Beispiel-B-Baum



Beispiel:

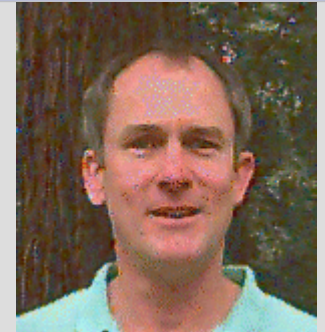
Index = x-Achse

Indexstrukturen

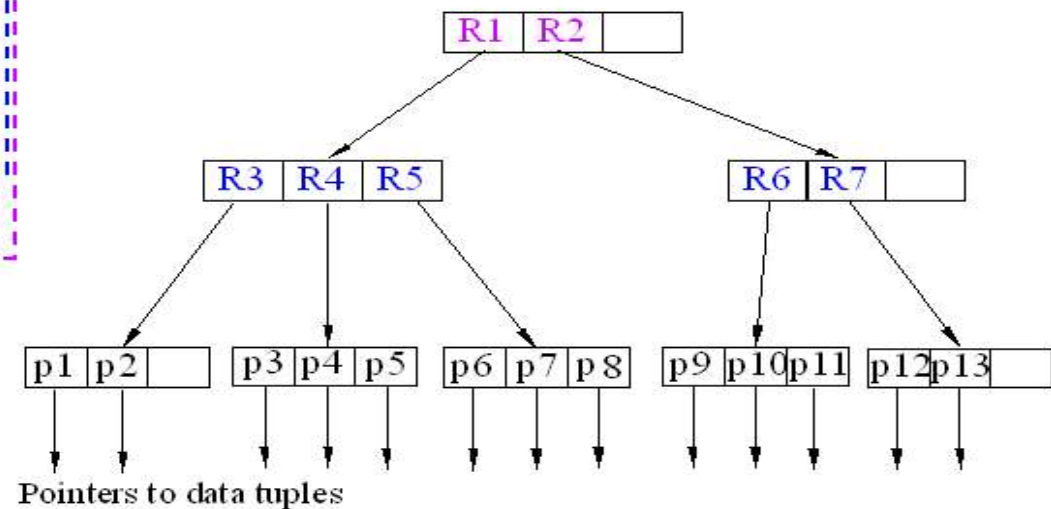
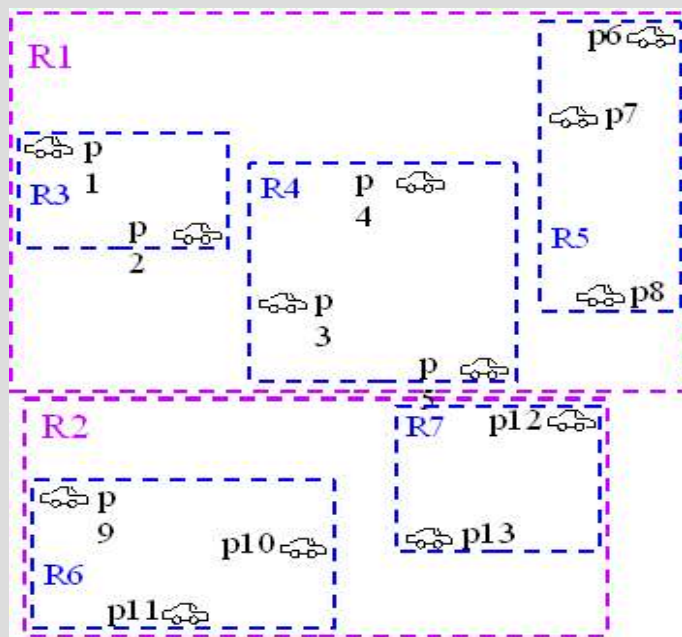
- B-Baum
 - keine effizienten räumlichen Suchanfragen möglich
- R-Baum
 - Erweiterung des B-Baum (mehrdimensionale Indexierung)
- R*-Baum
 - „Performance-optimierter“ R-Baum
- TPR-Baum
 - Anpassung des R*-Baum für „sich stetig fortbewegende Objekte“

R-Baum

Antonin Guttman, 1984

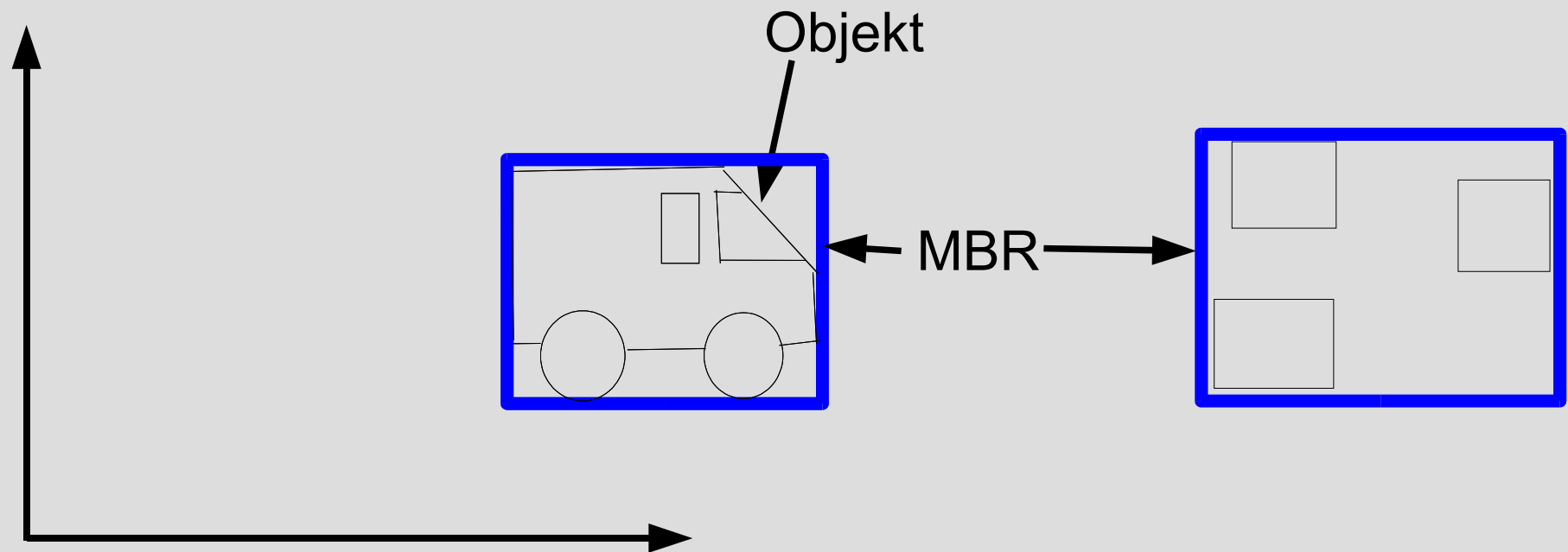


- „B⁺-Baum“ für räumliche Daten
- Koordinaten als Indizes



Minimum Bounding Rectangle (MBR)

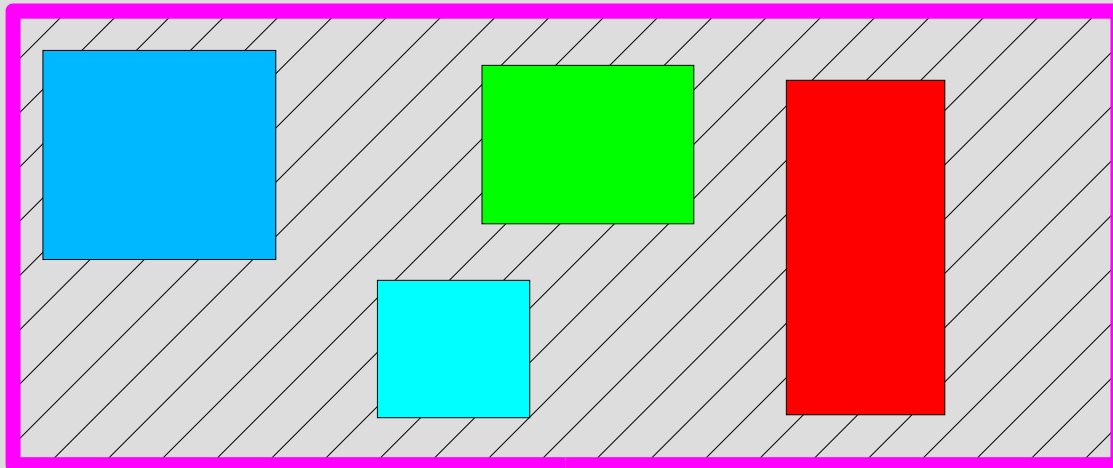
- kleinstmögliches Rechteck
- umschließt Objekt / enthaltene Rechtecke
- parallel zu Koordinatenachsen



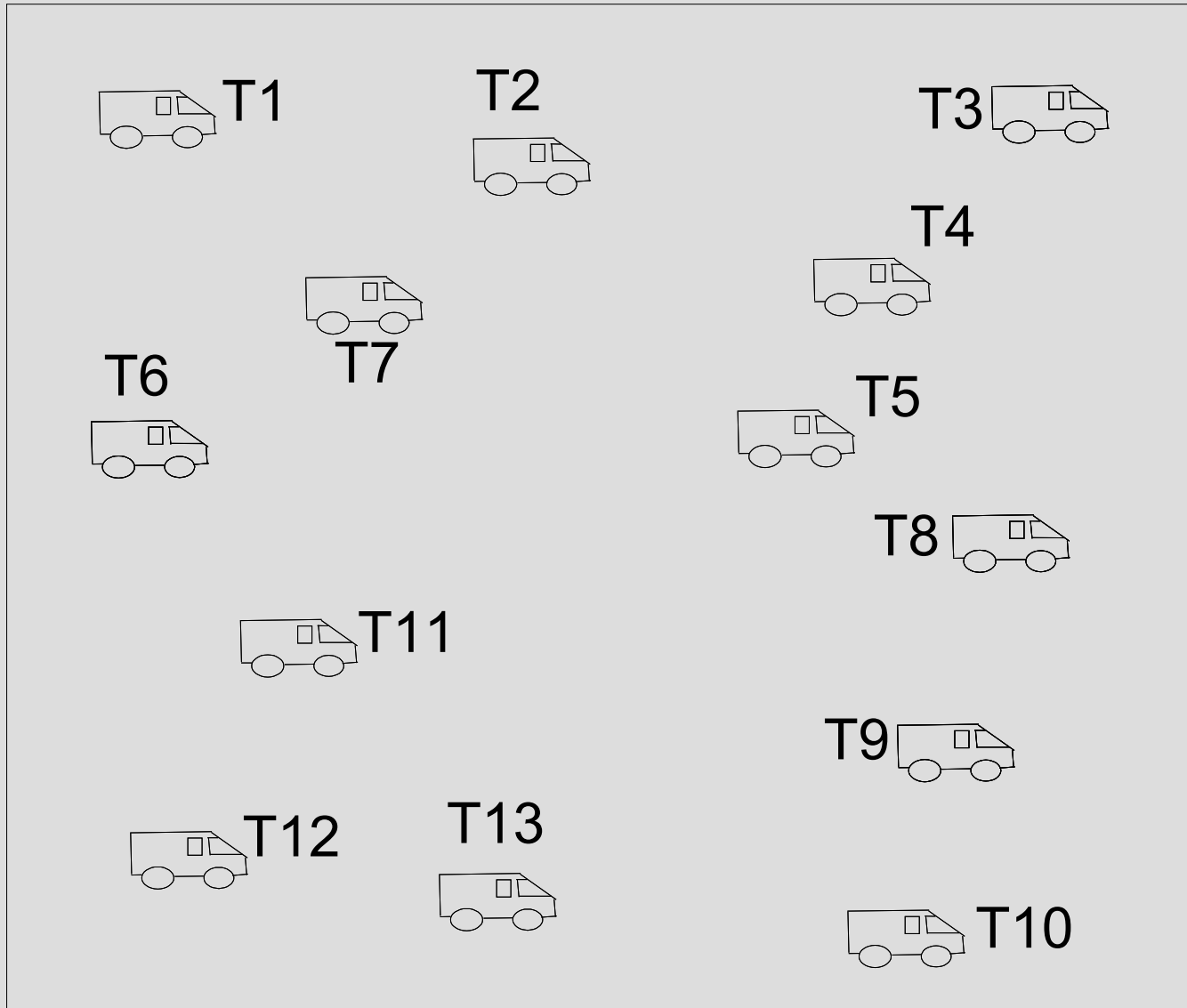
Ordnung

Nach welchem Kriterien sollen die Knoten geordnet werden?

- **Minimiere Fläche** zwischen einem MBR und den darin enthaltenen Rechtecken.



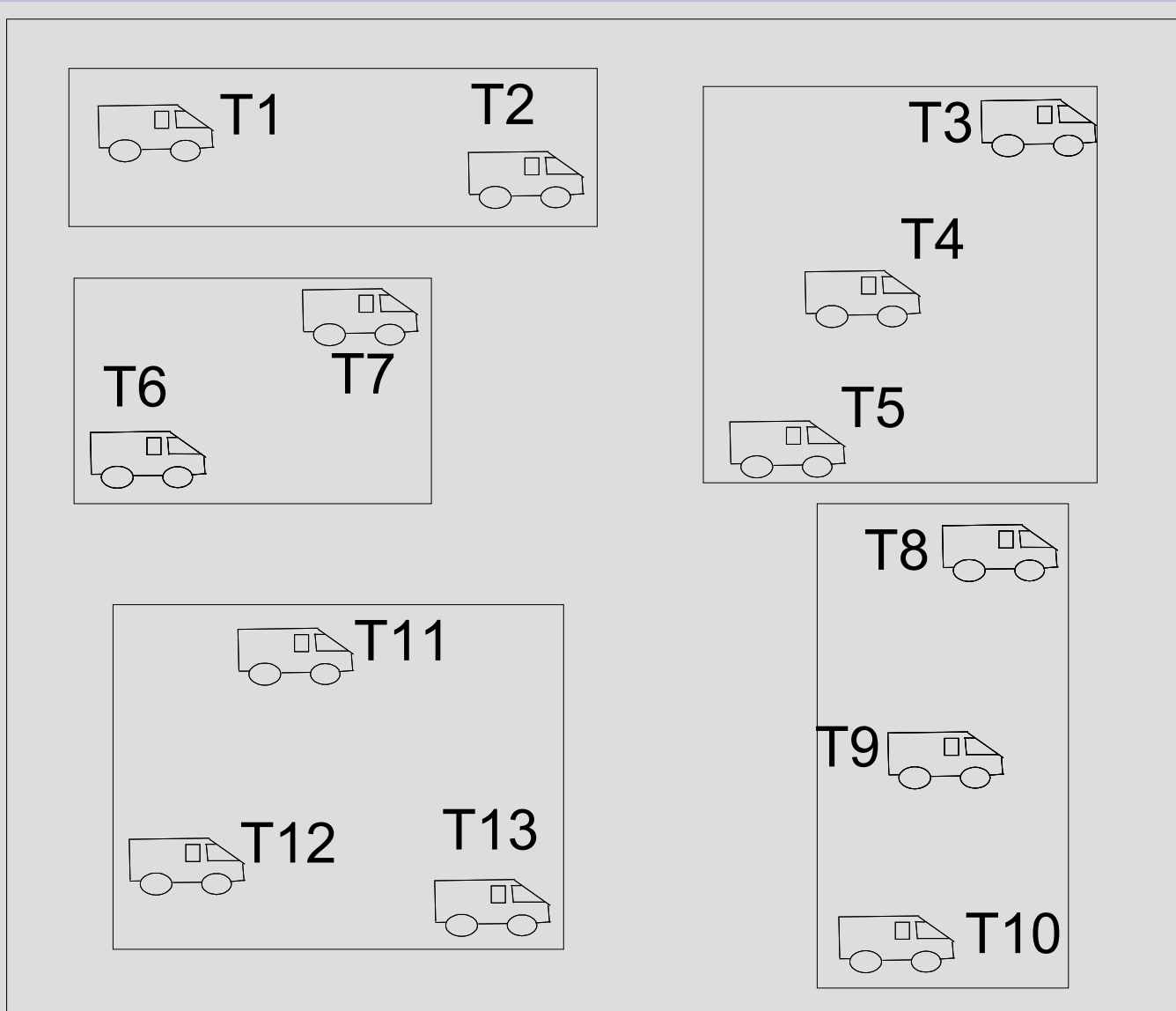
Beispiel-R-Baum



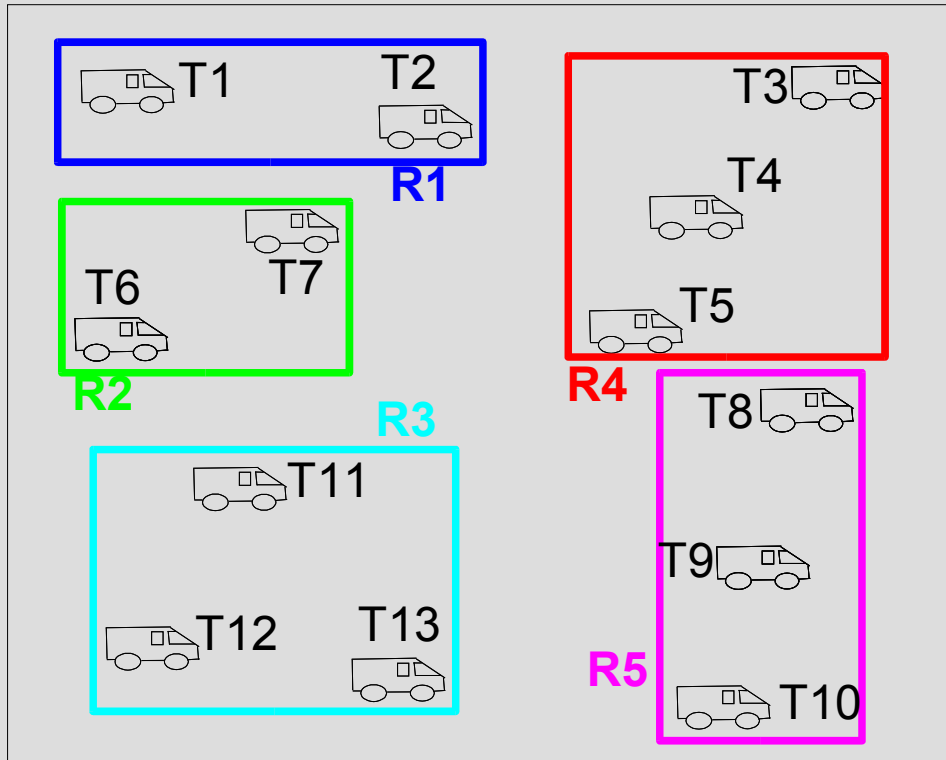
Annahme:

3 Elemente/Knoten

Beispiel-R-Baum



Beispiel-R-Baum



T1 T2

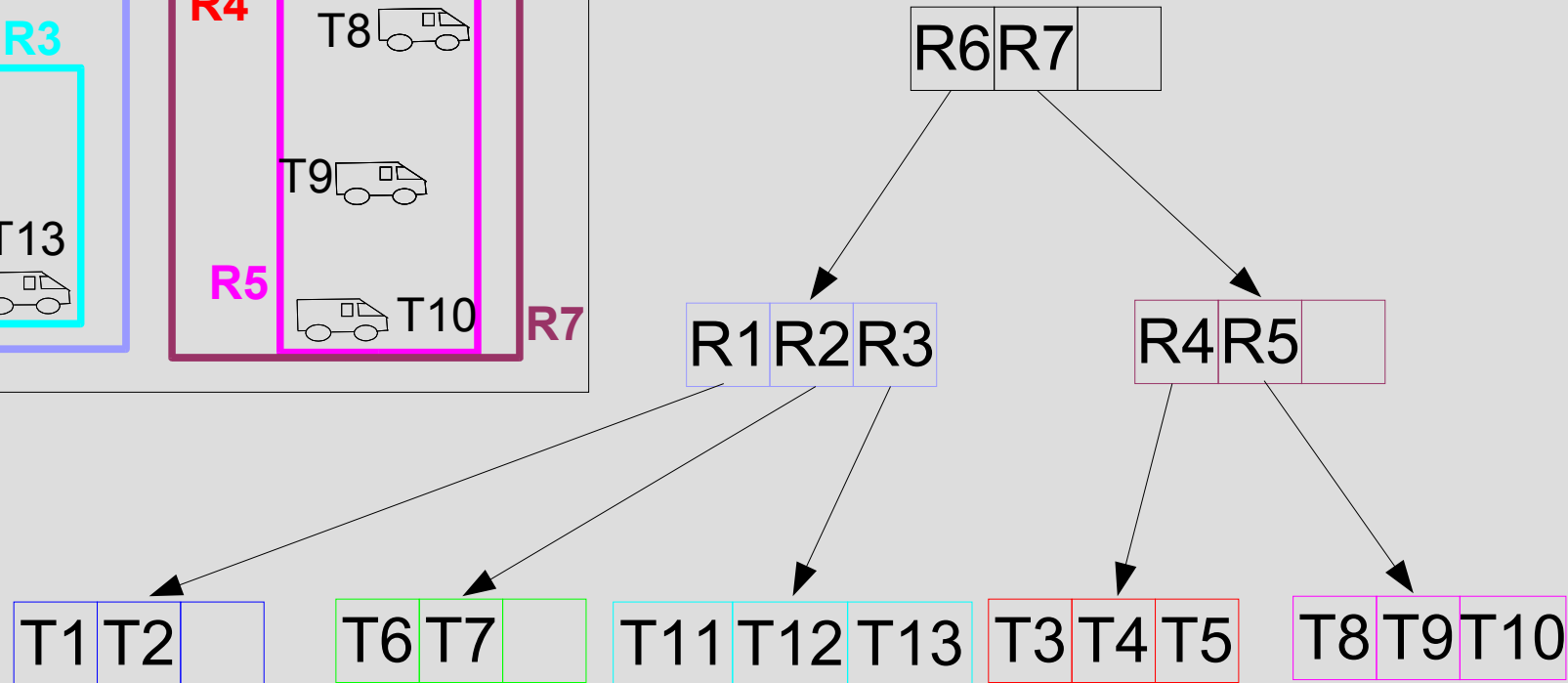
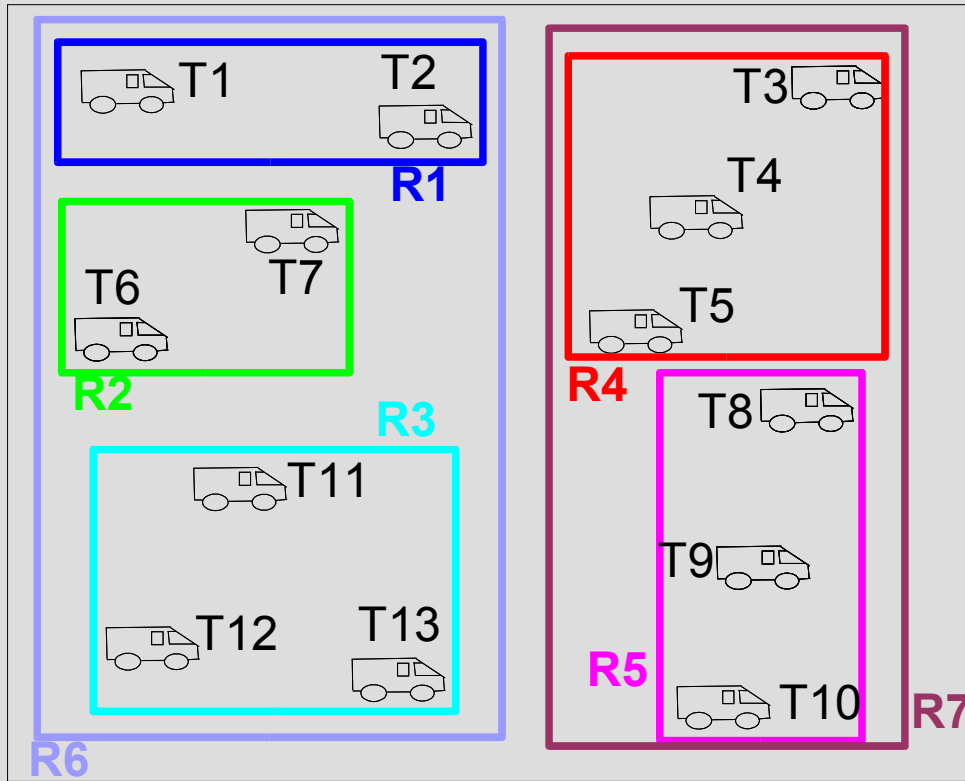
T3 T4 T5

T6 T7

T8 T9 T10

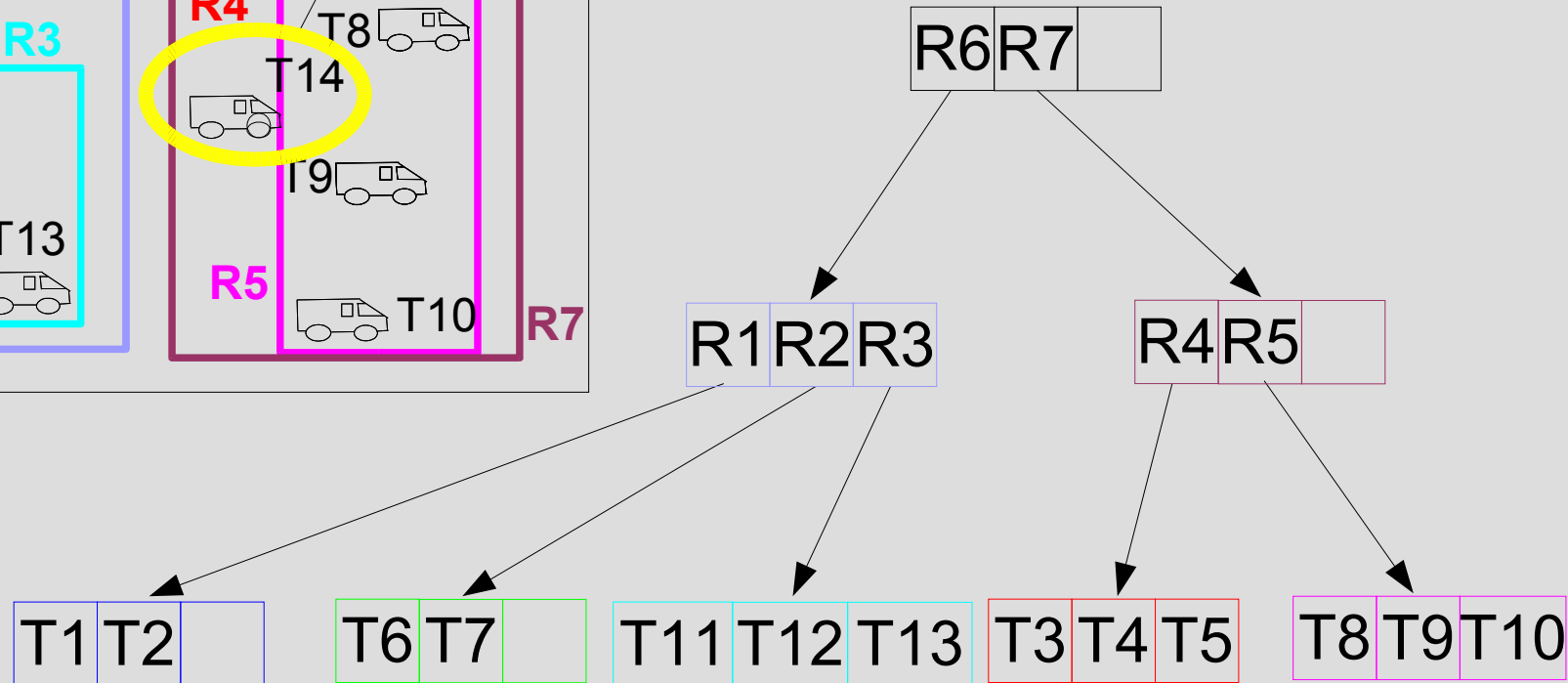
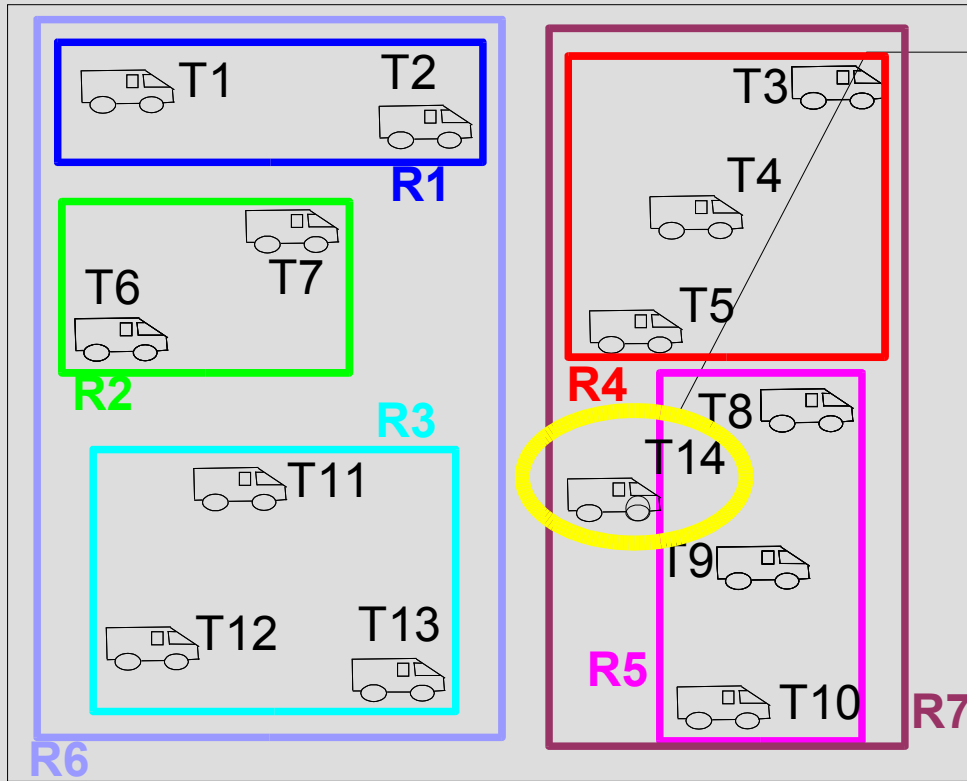
T11 T12 T13

Beispiel-R-Baum



Insert

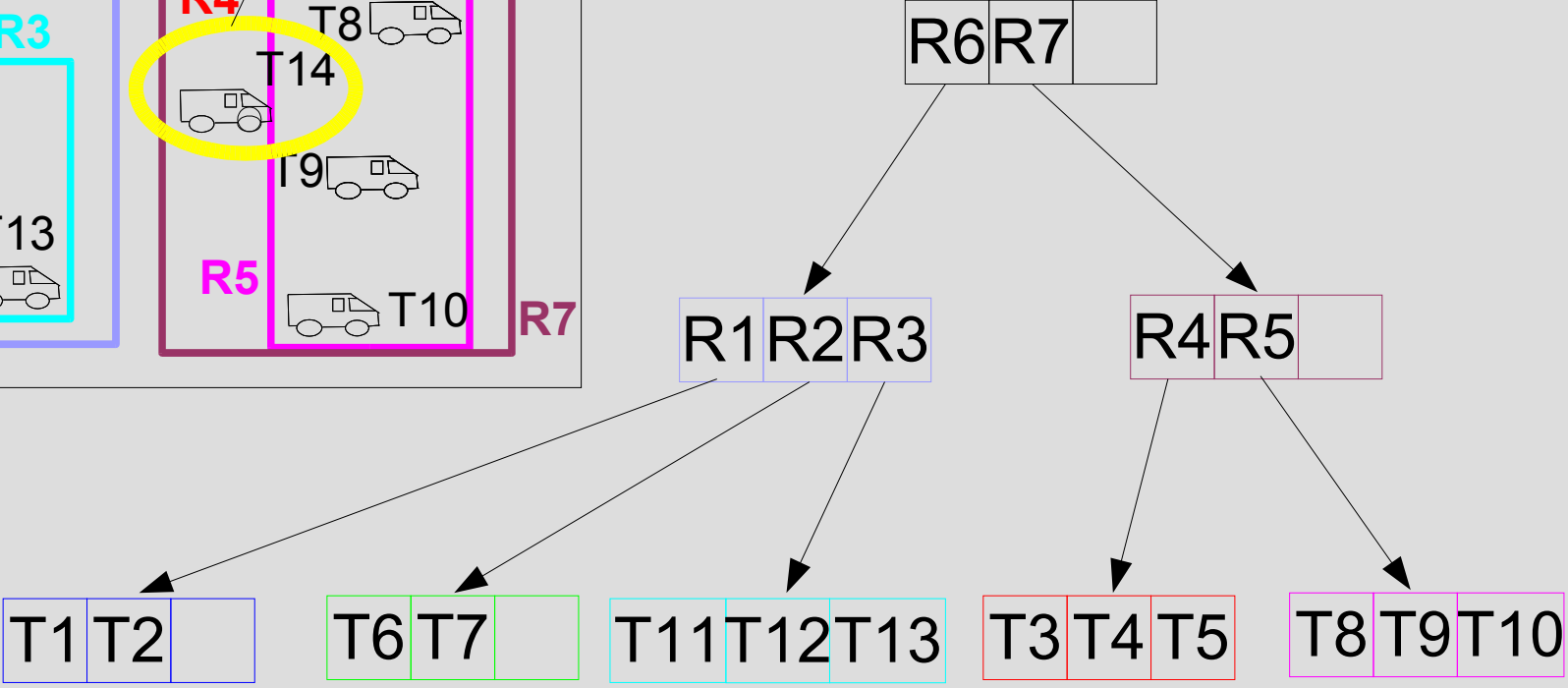
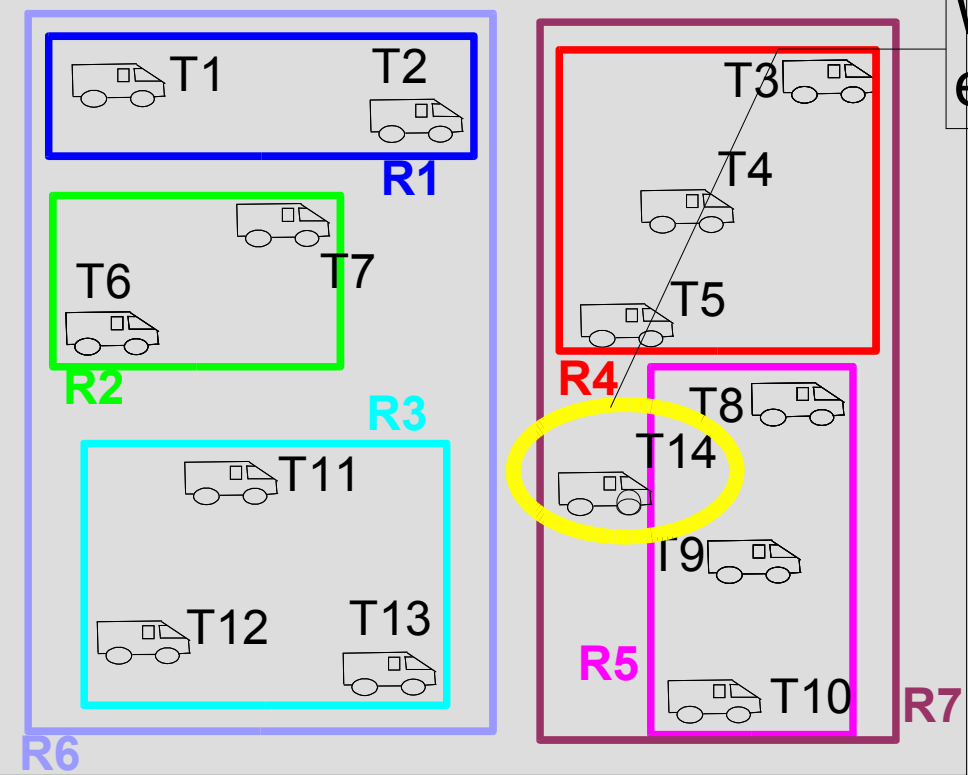
Wo soll der neue Transporter eingefügt werden?



Insert

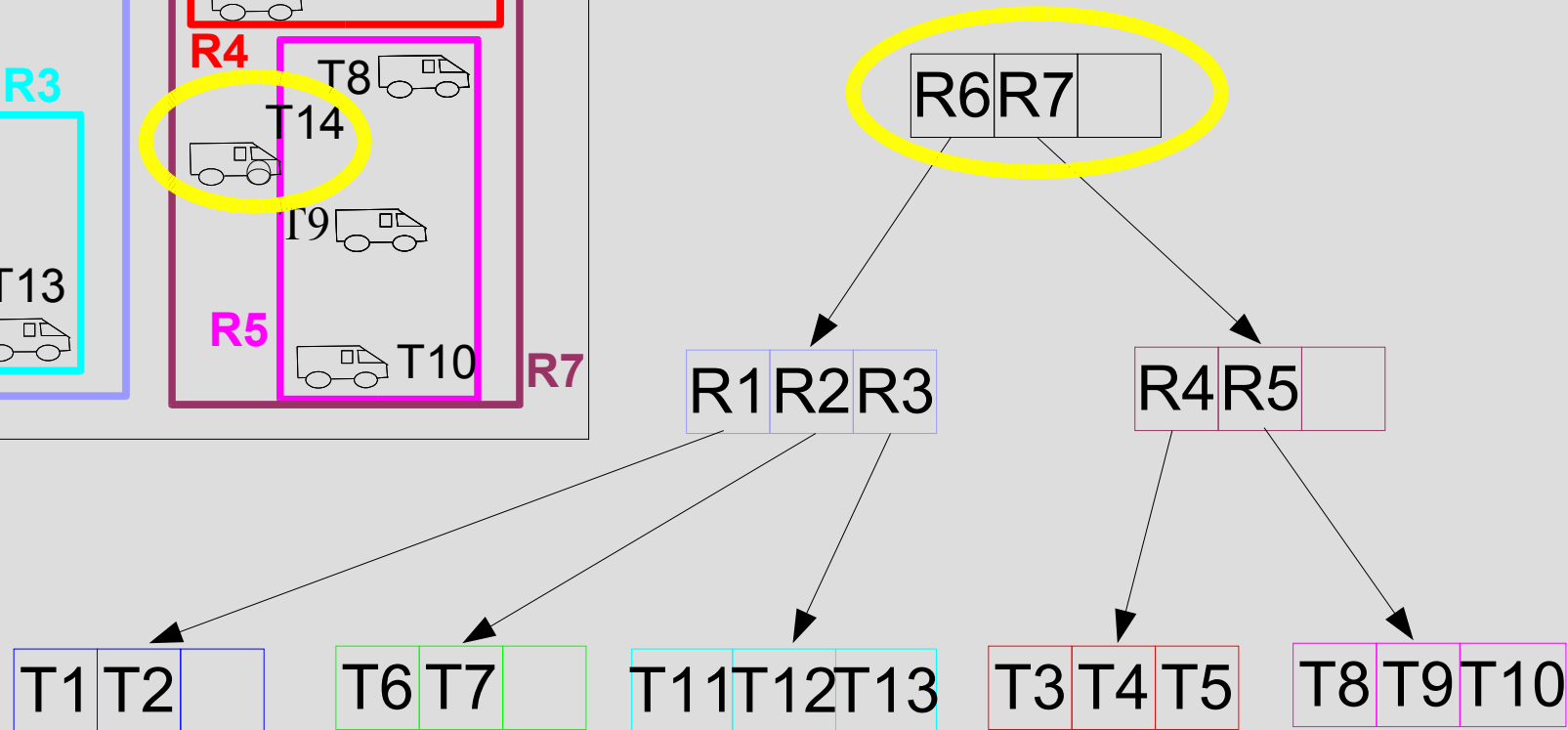
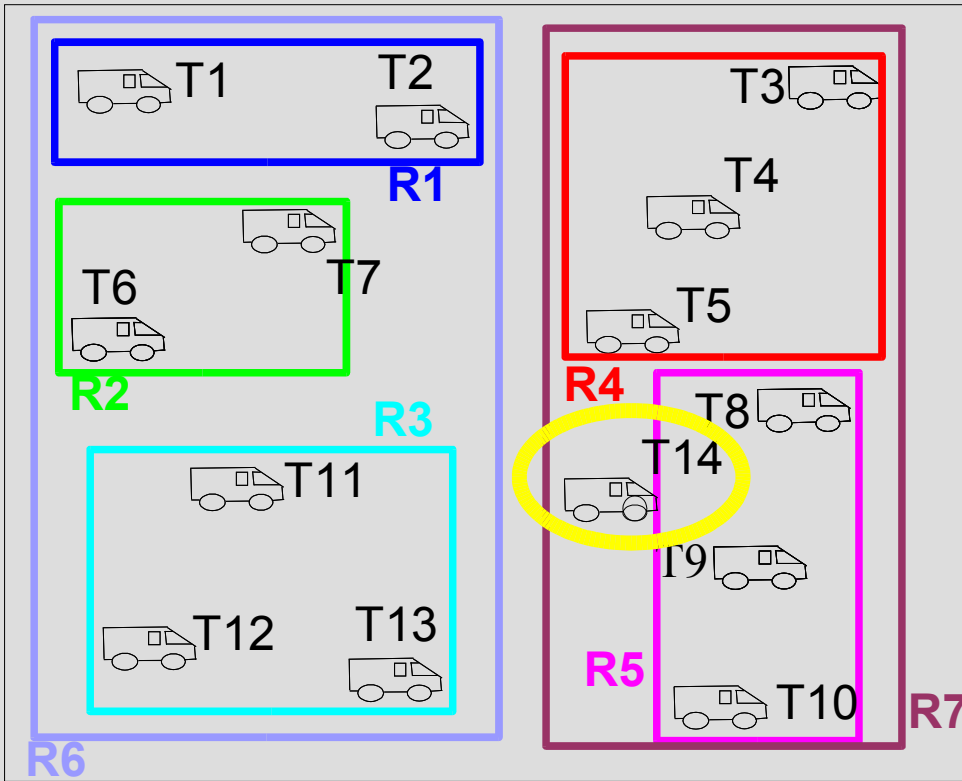
Wo soll der neue Transporter eingefügt werden?

ChooseLeaf

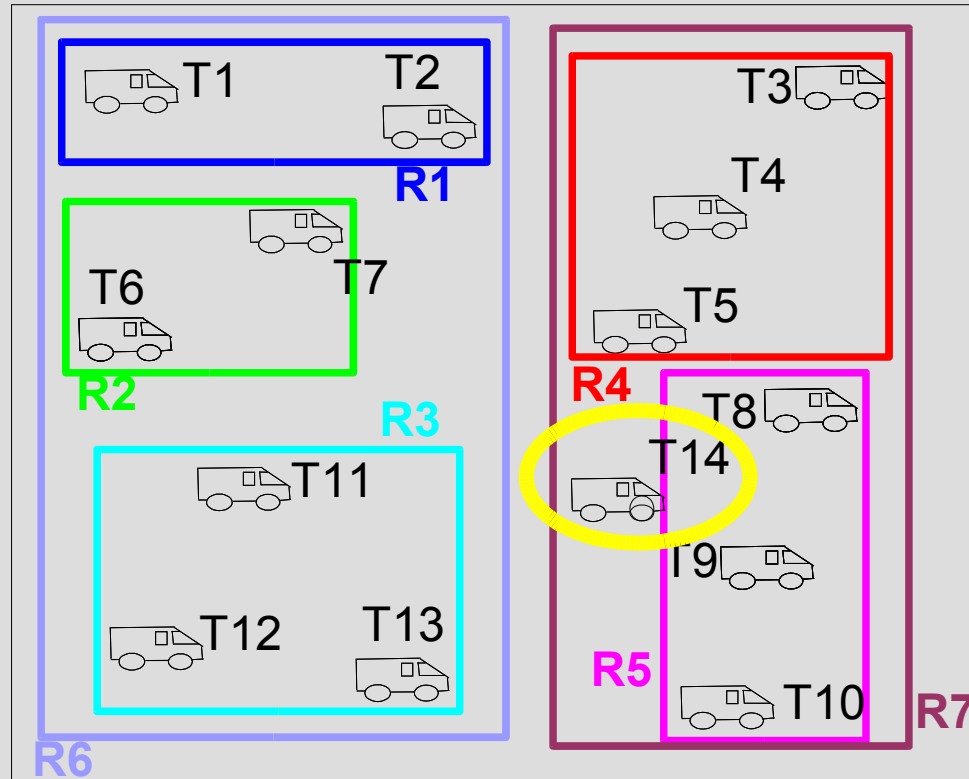


Insert - ChooseLeaf

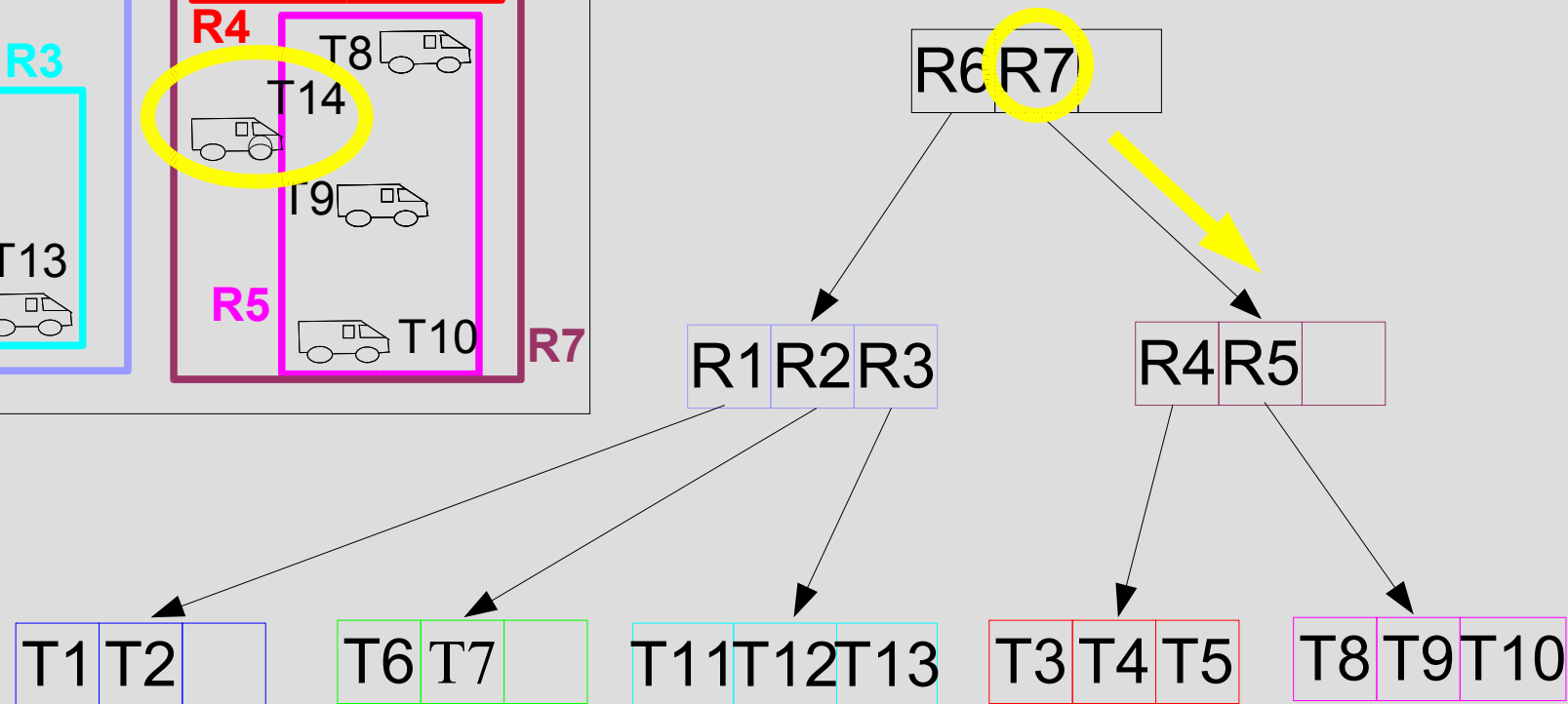
Kleinste Flächenvergrößerung?



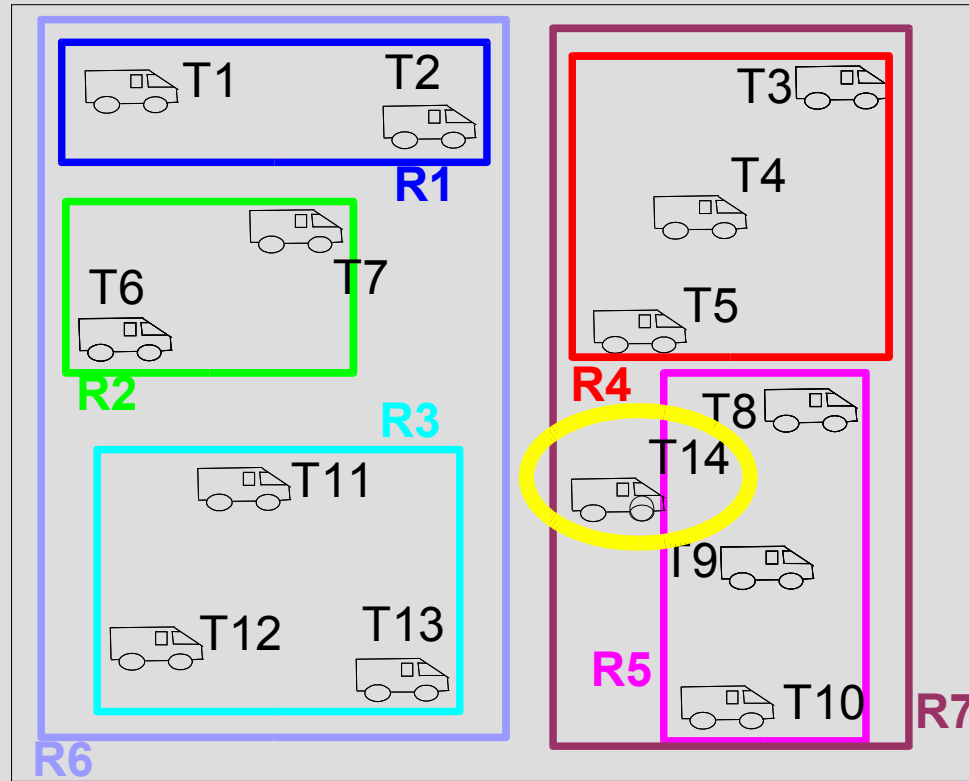
Insert - ChooseLeaf



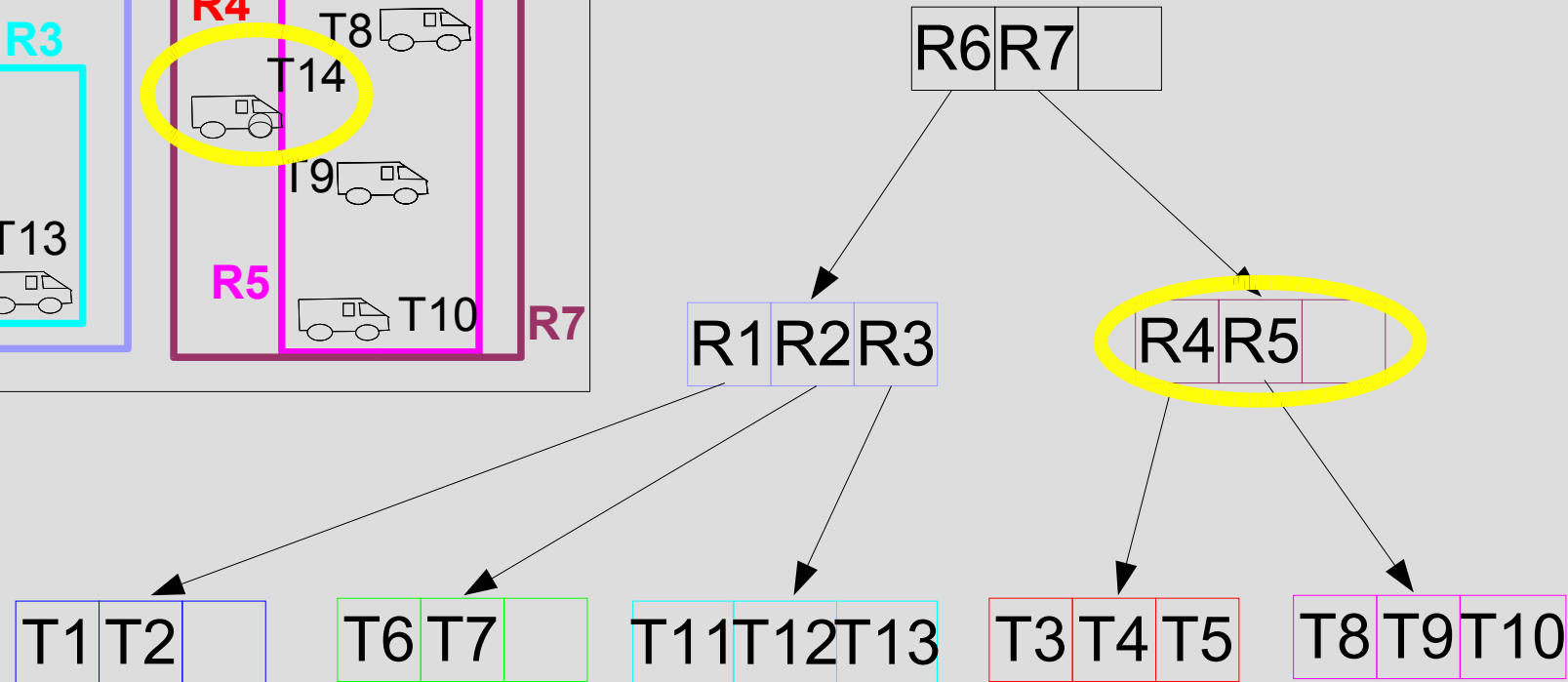
R7 benötigt keine!



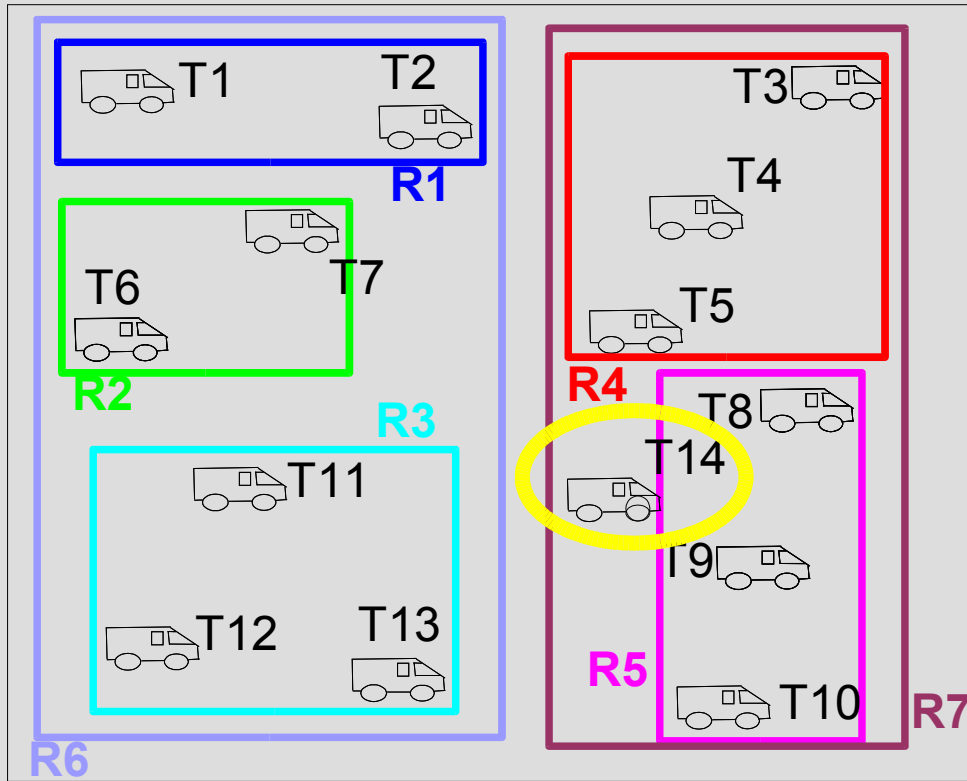
Insert - ChooseLeaf



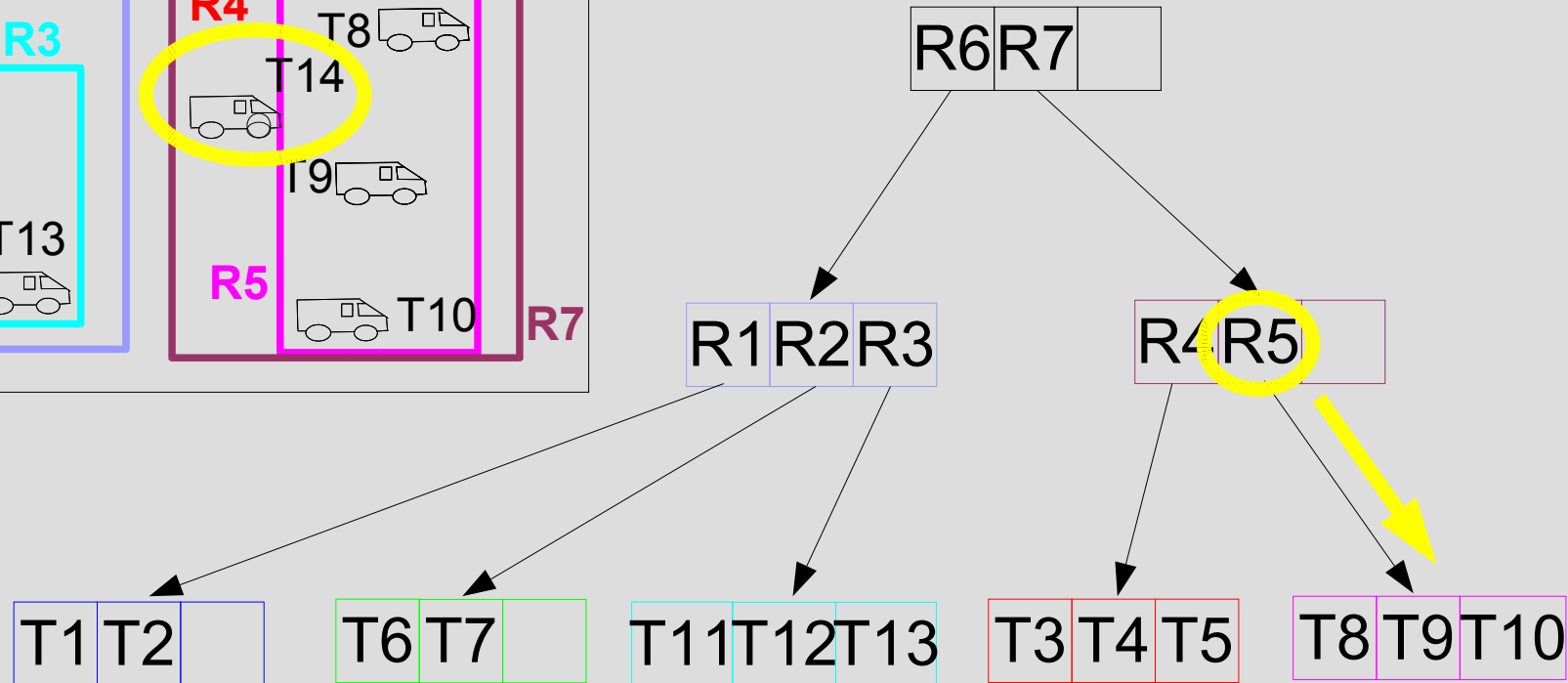
Kleinste Flächenvergrößerung?



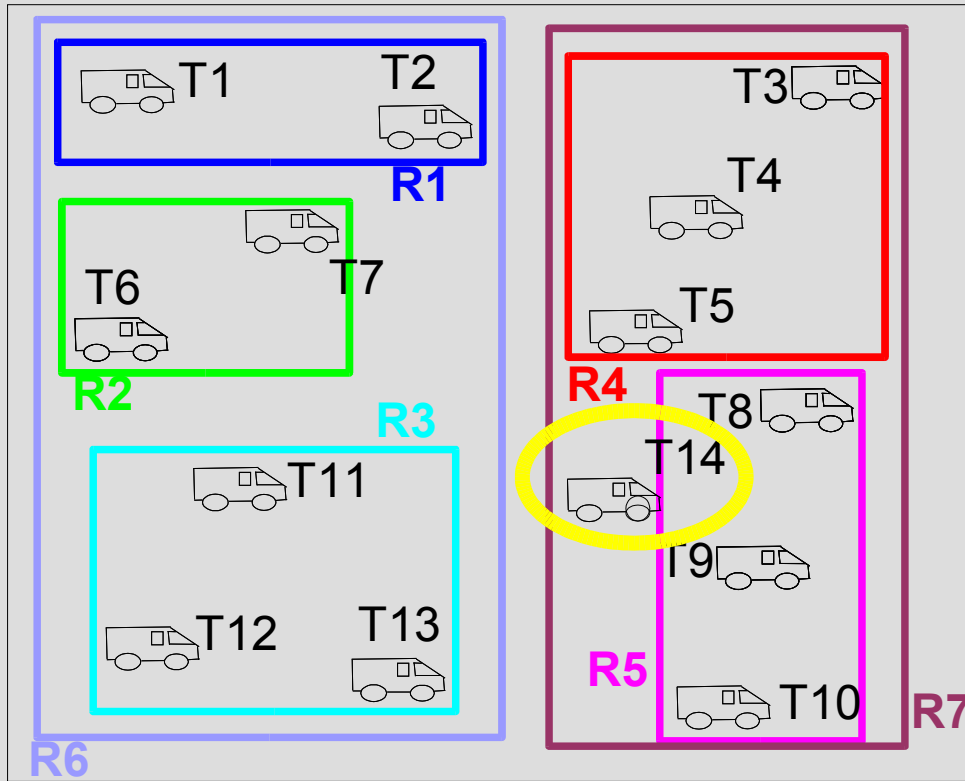
Insert - ChooseLeaf



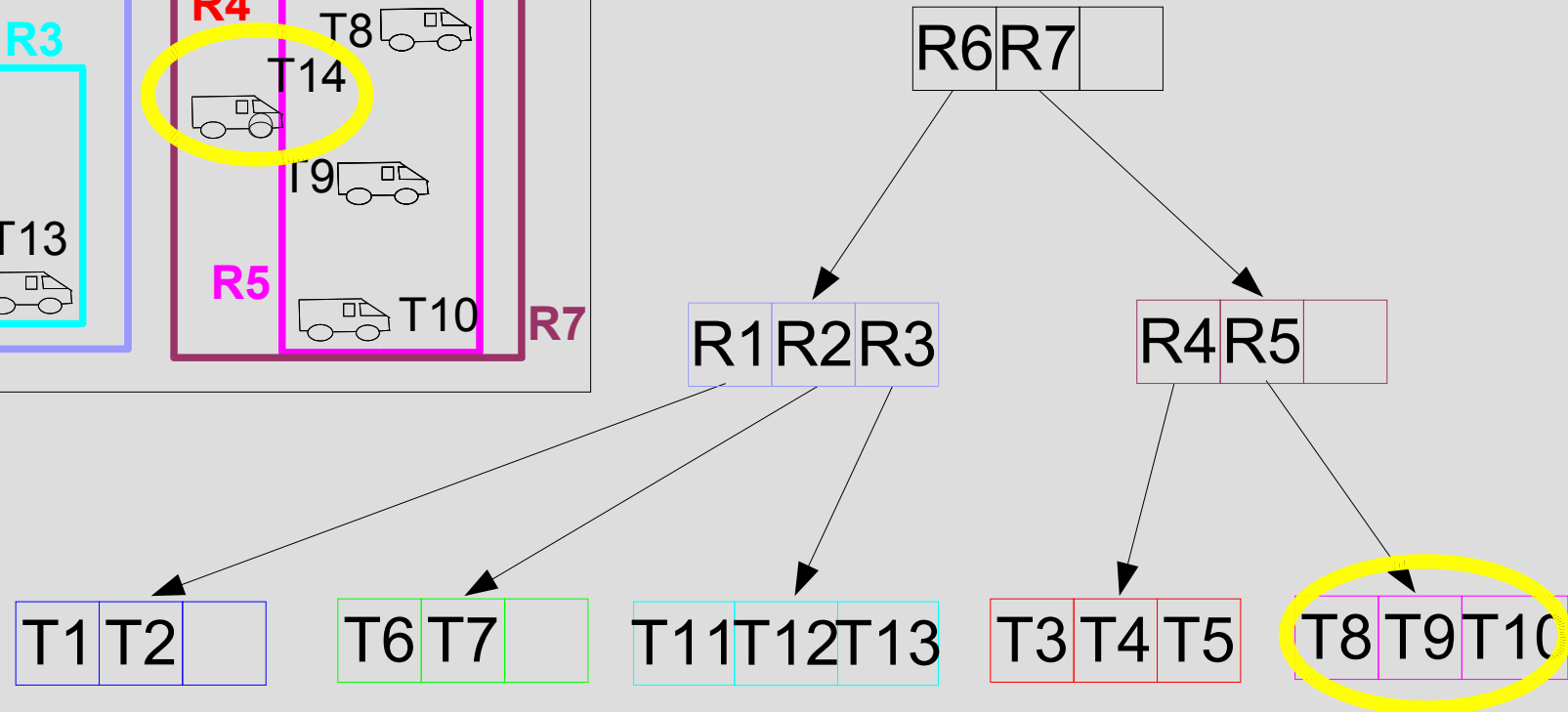
R5!



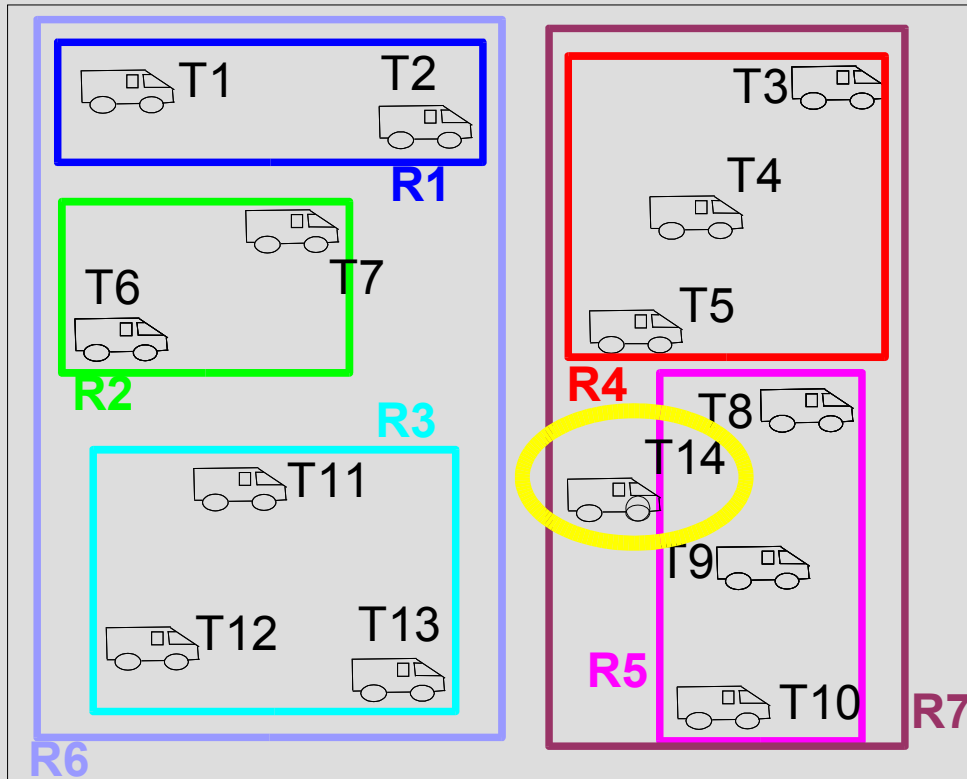
Insert - ChooseLeaf



Knoten gefunden!

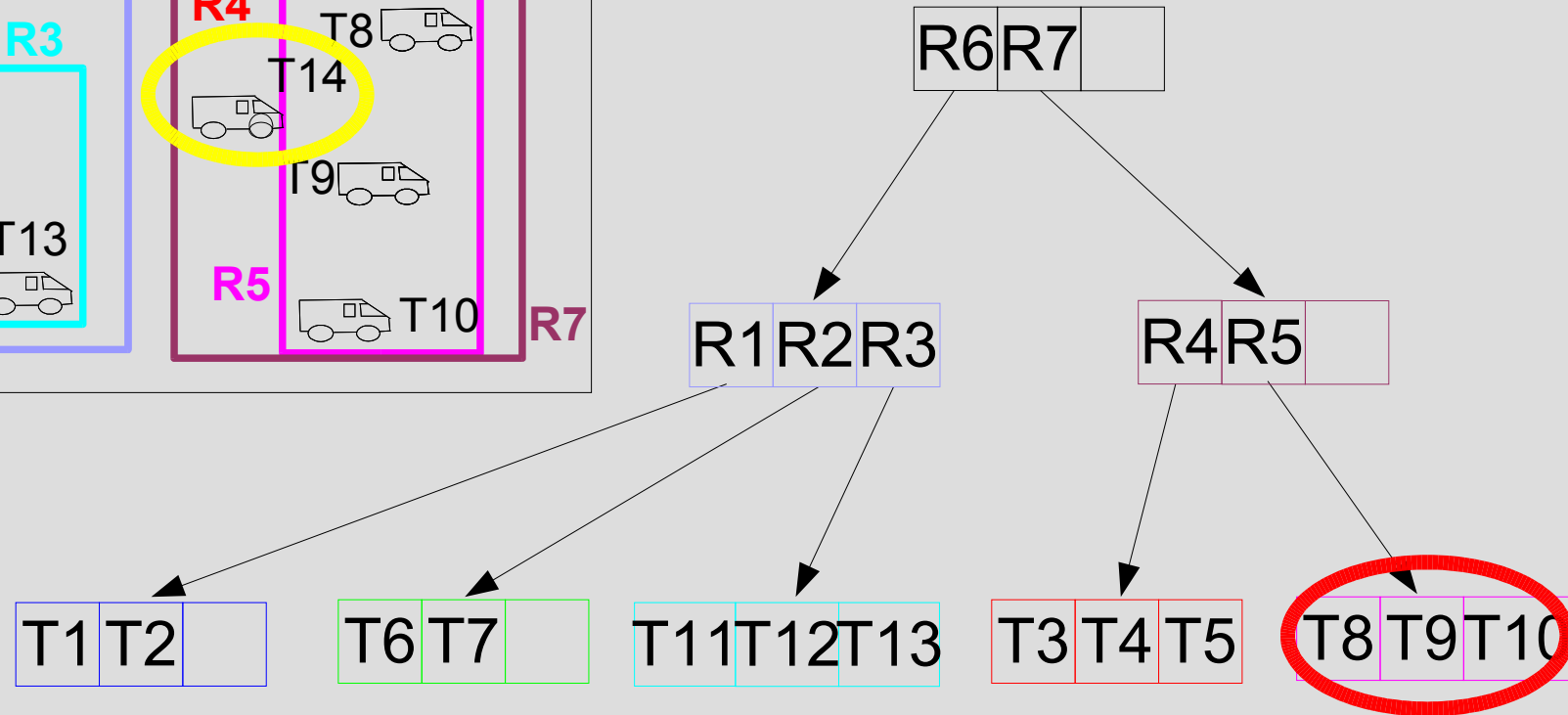


Insert

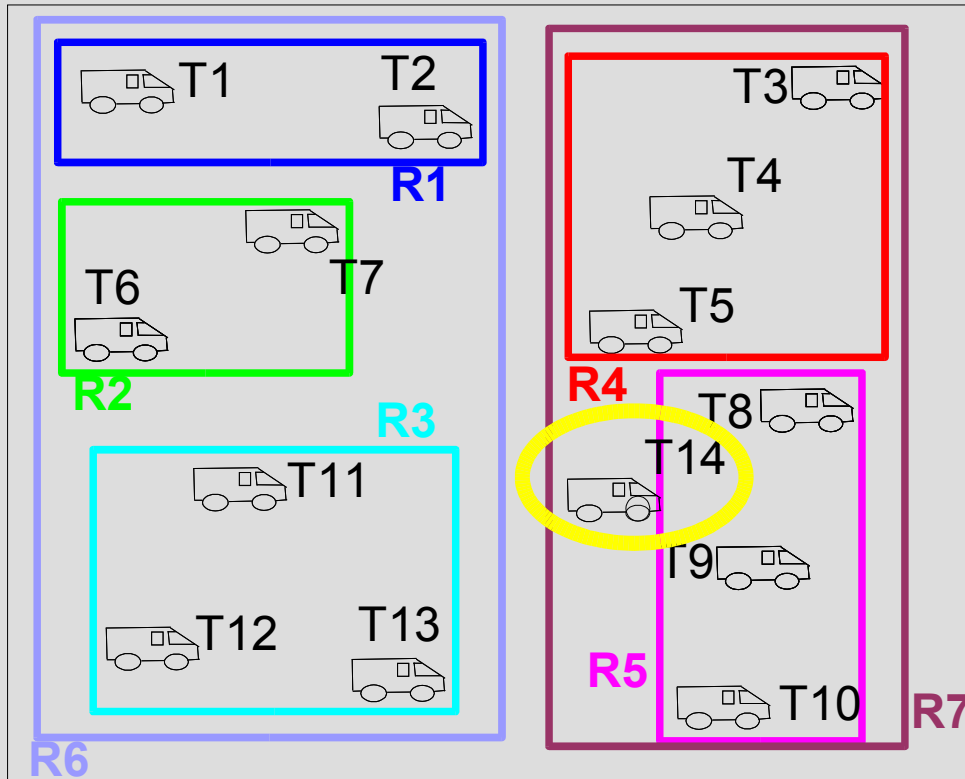


Knoten voll!!!

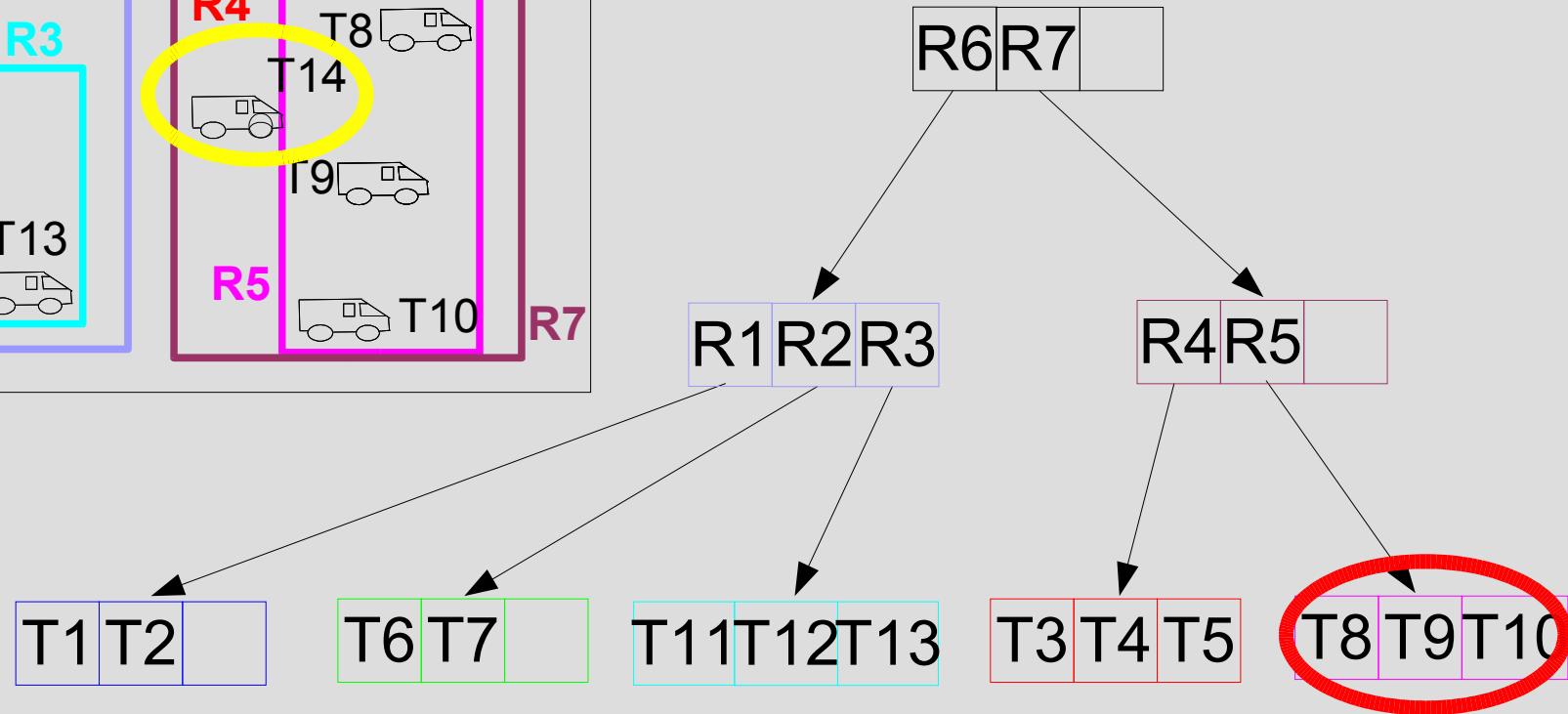
=> **NodeSplit**



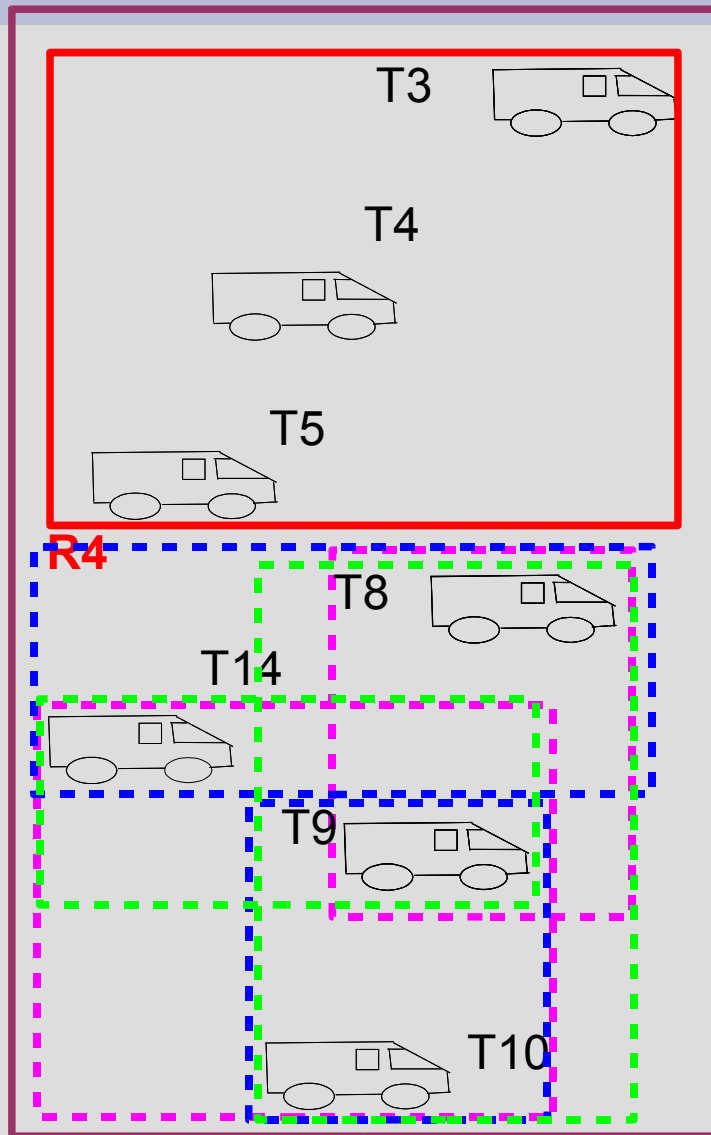
Insert - NodeSplit



Exhaustive
Quadratic Cost
Linear Cost



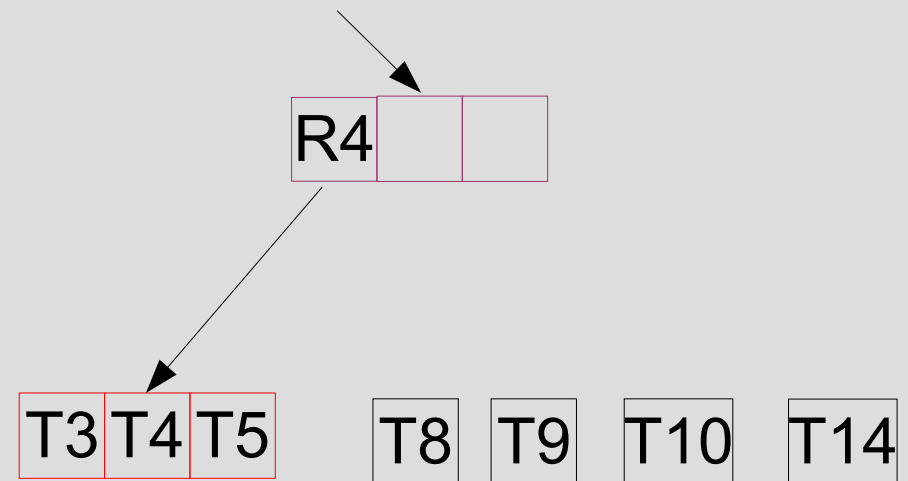
Insert - NodeSplit



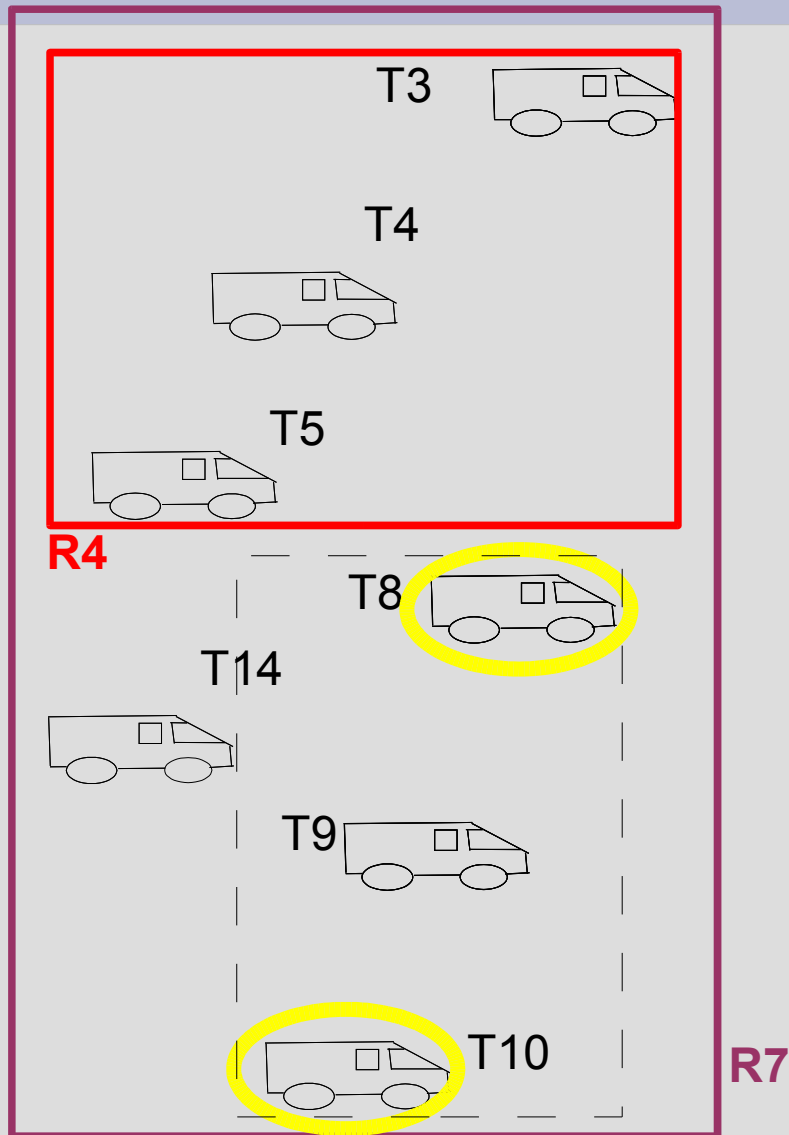
Exhaustive

Quadratic Cost
Linear Cost

Alle Möglichkeiten durchprobieren.
Beste wählen.

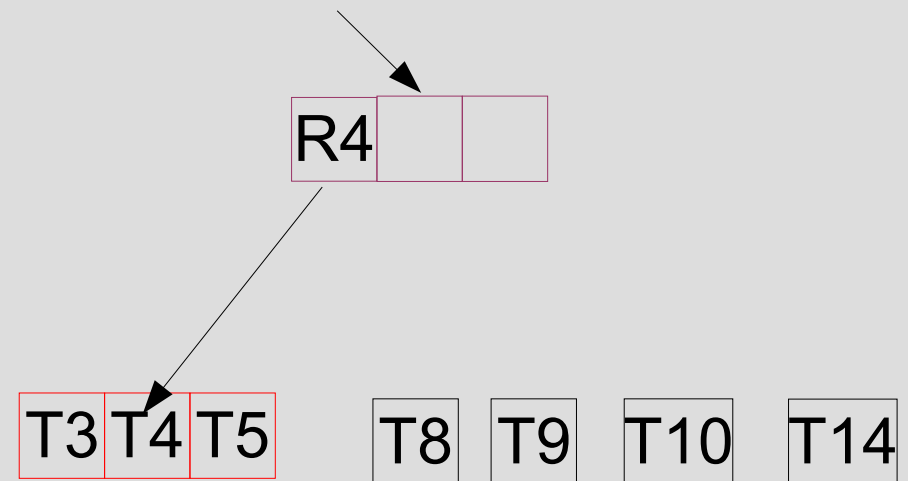


Insert - NodeSplit

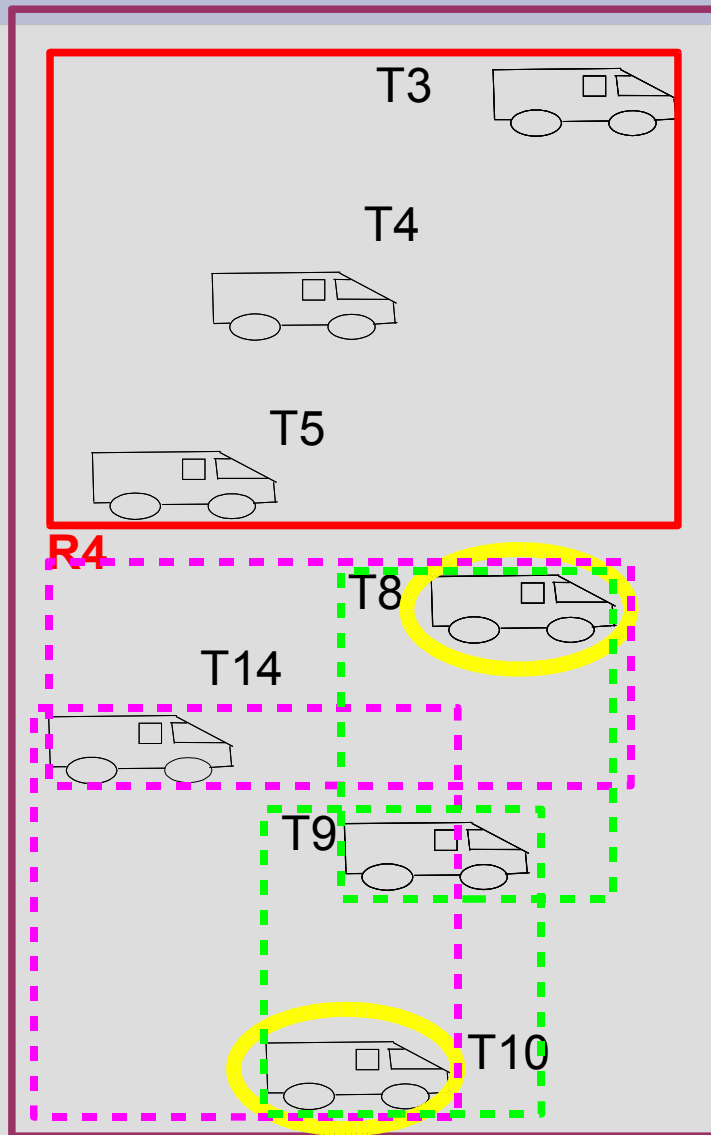


Exhaustive
Quadratic Cost
Linear Cost

Paarweise Fläche berechnen.
Paar mit größter Fläche wählen.

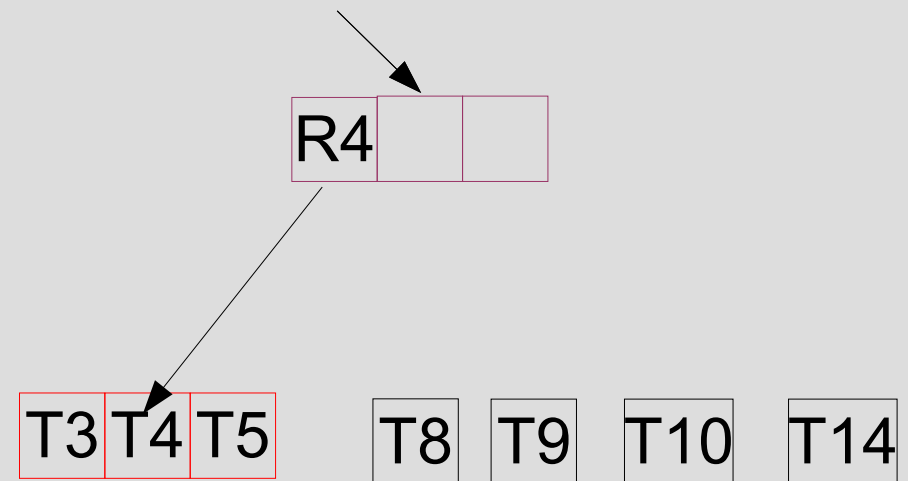


Insert - NodeSplit

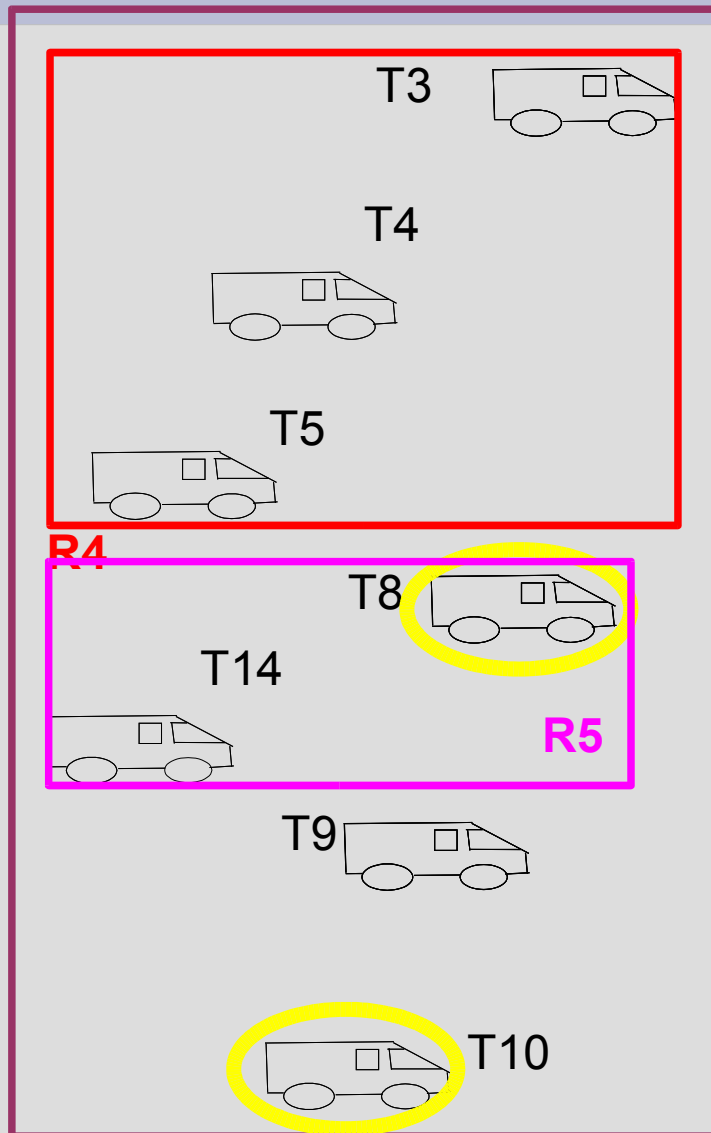


Exhaustive
Quadratic Cost
Linear Cost

Für alle übrigen Einträge:
-berechne Flächen mit gewählten Einträgen
-wähle Eintrag mit gr. Differenz der Flächen

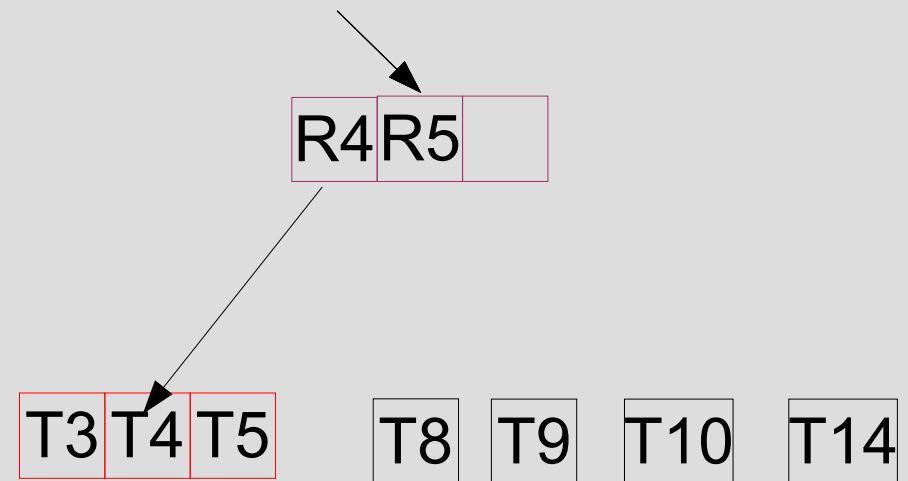


Insert - NodeSplit

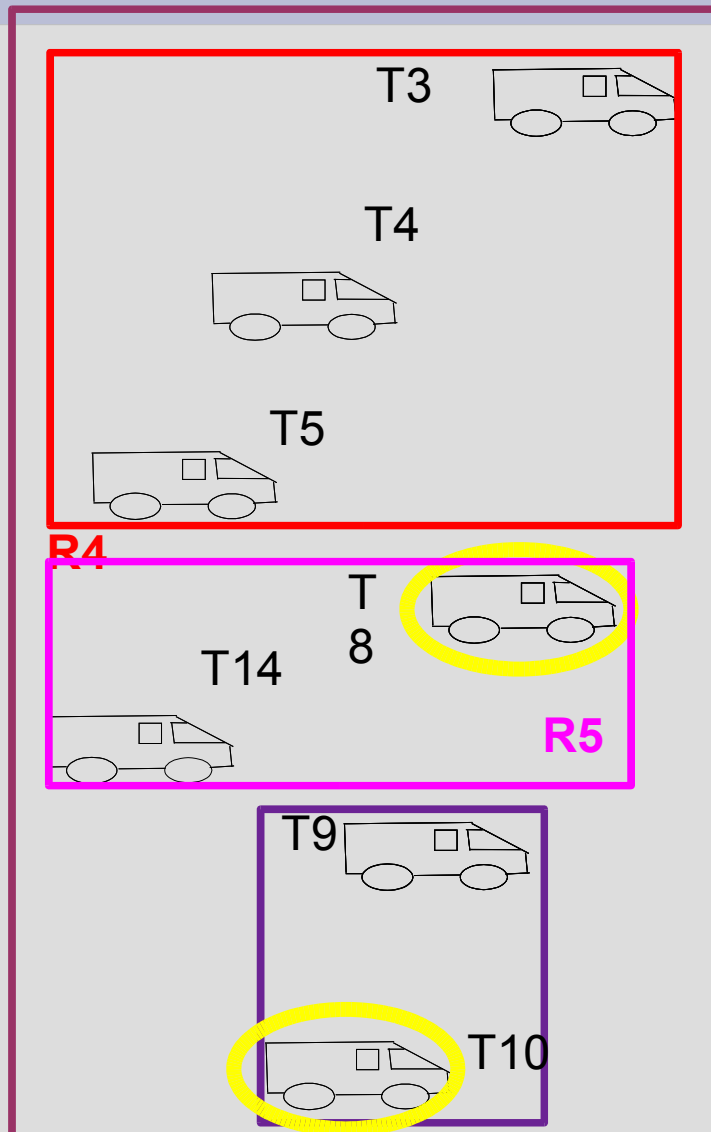


Exhaustive
Quadratic Cost
Linear Cost

Für alle übrigen Einträge:
-berechne Flächen mit gewählten Einträgen
-wähle Eintrag mit gr. Differenz der Flächen
-füge ihn zu Rechteck, das am wenigsten vergrößert werden muss

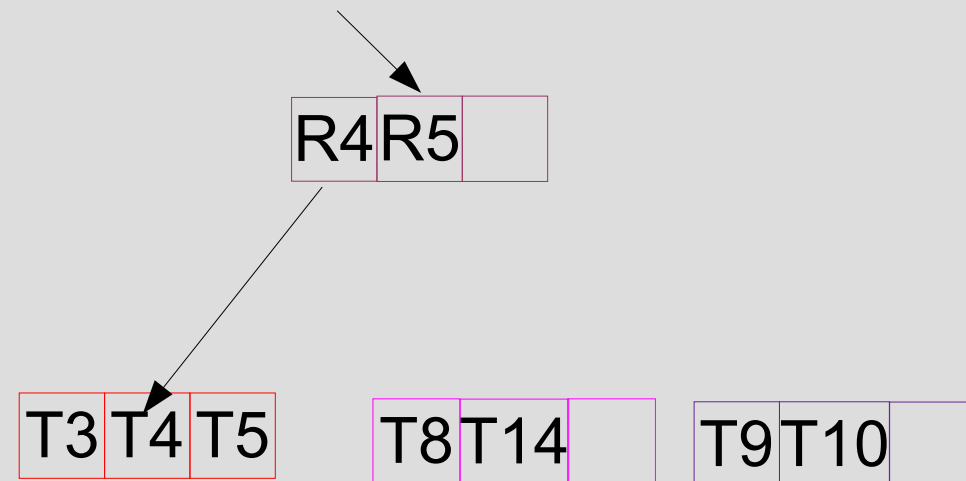


Insert - NodeSplit

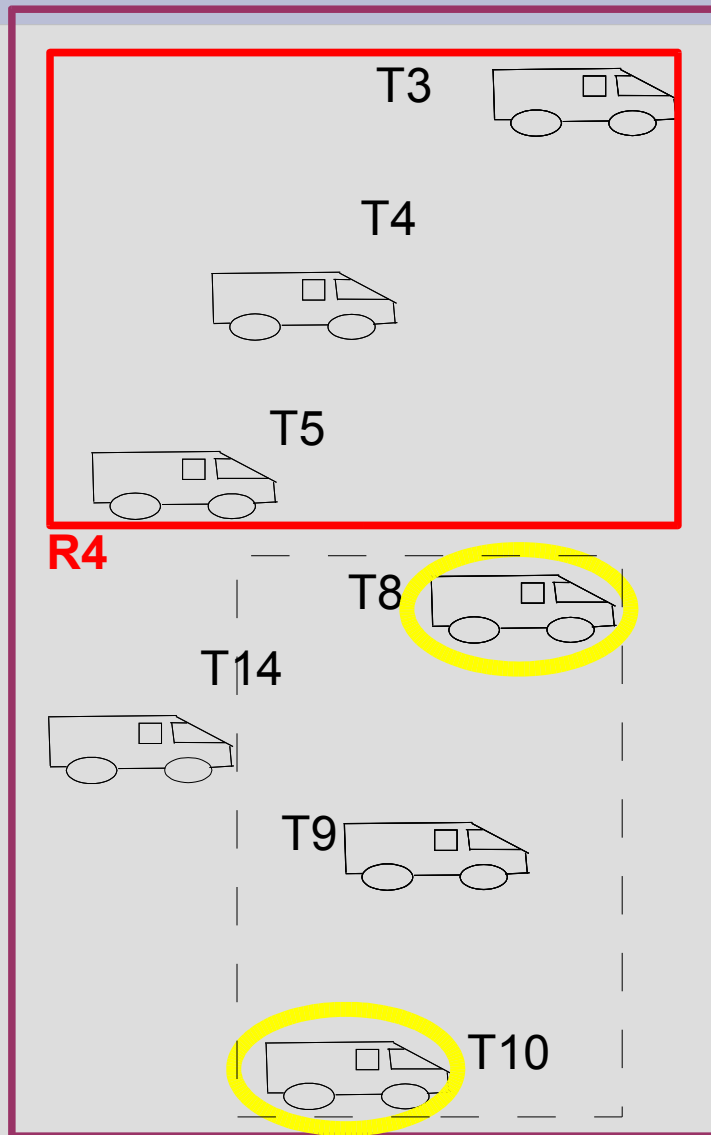


Exhaustive
Quadratic Cost
Linear Cost

Für alle übrigen Einträge:
-berechne Flächen mit gewählten Einträgen
-wähle Eintrag mit gr. Differenz der Flächen
-füge ihn zu Rechteck, das am wenigsten vergrößert werden muss



Insert - NodeSplit



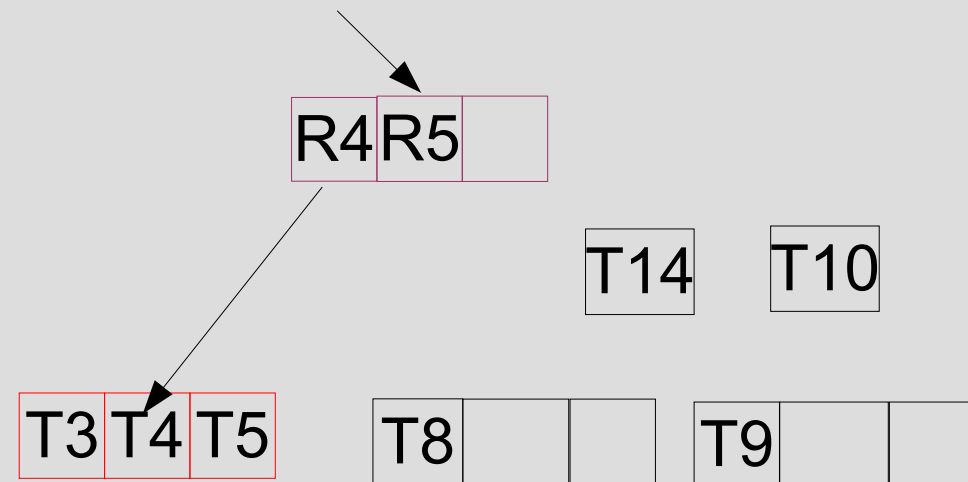
Exhaustive
Quadratic Cost

Linear Cost

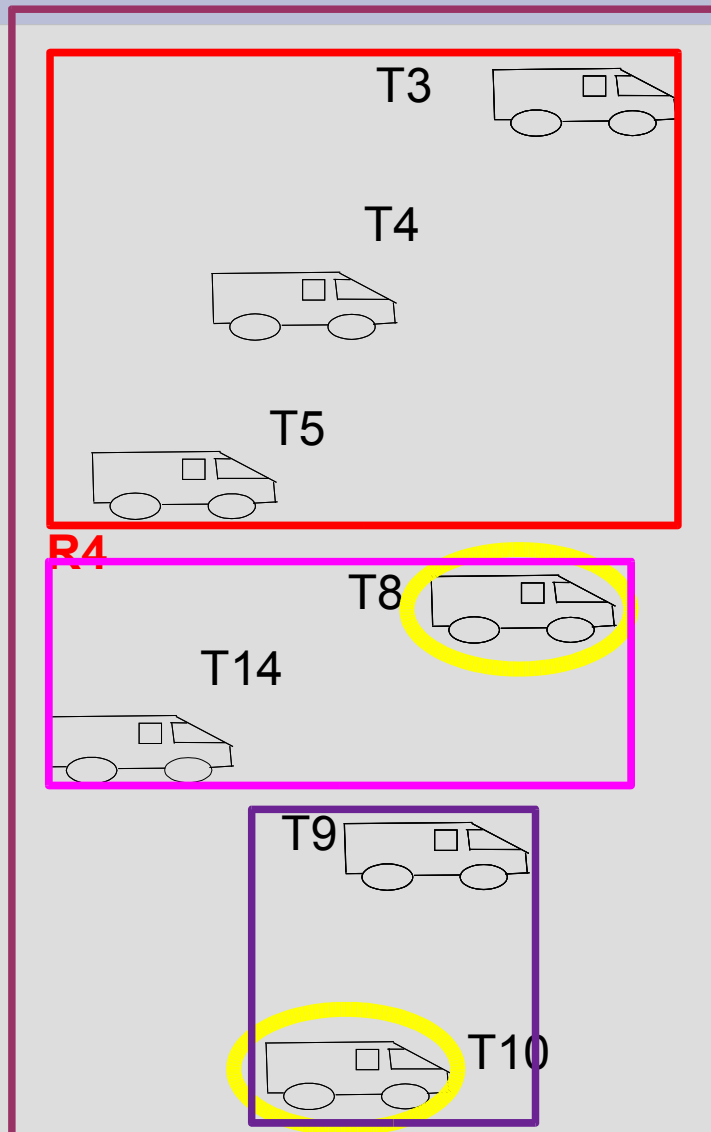
Wie Quadratic Cost.

Unterschied:

Nachdem 2 Starteinträge gefunden wurden, wird der Rest zufällig eingefügt



Insert - NodeSplit



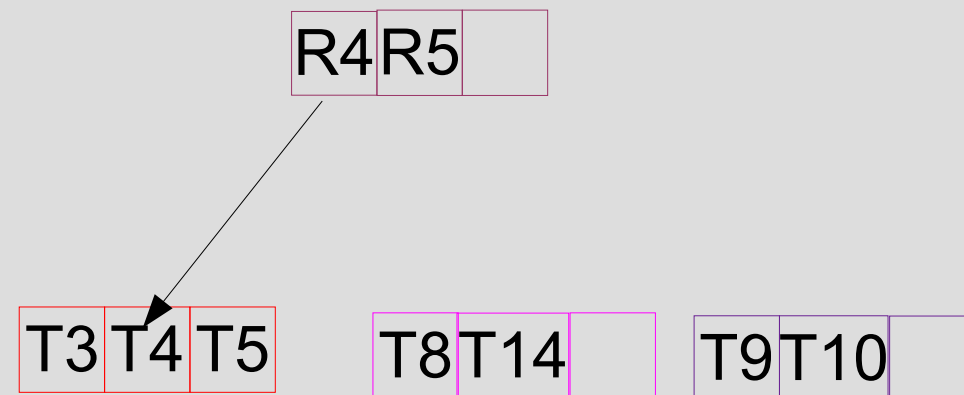
Exhaustive
Quadratic Cost

Linear Cost

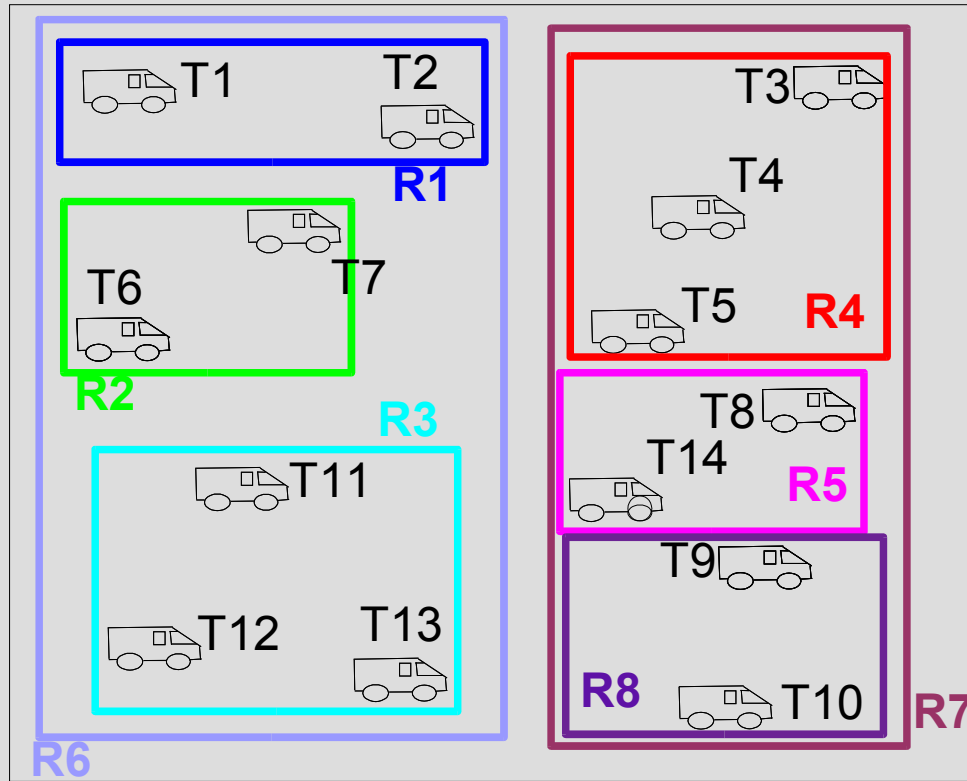
Wie Quadratic Cost.

Unterschied:

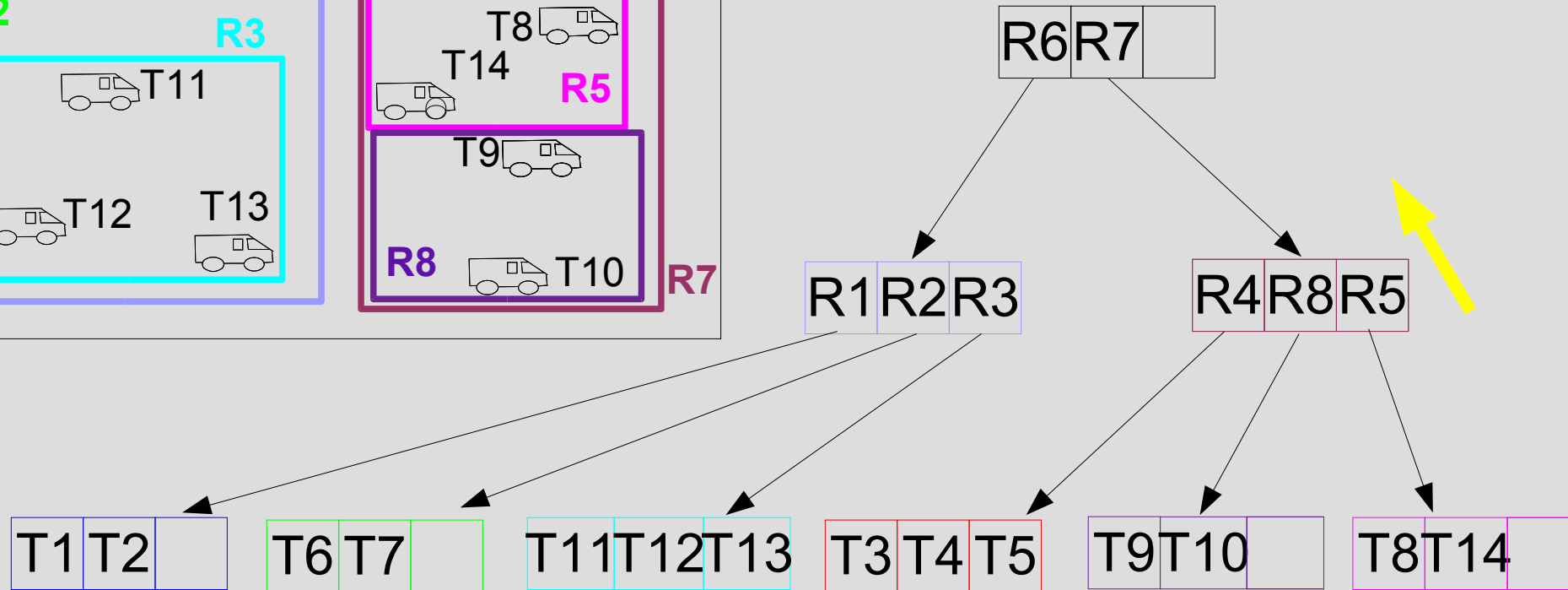
Nachdem 2 Starteinträge gefunden wurden, wird der Rest zufällig eingefügt



Insert – AdjustTree

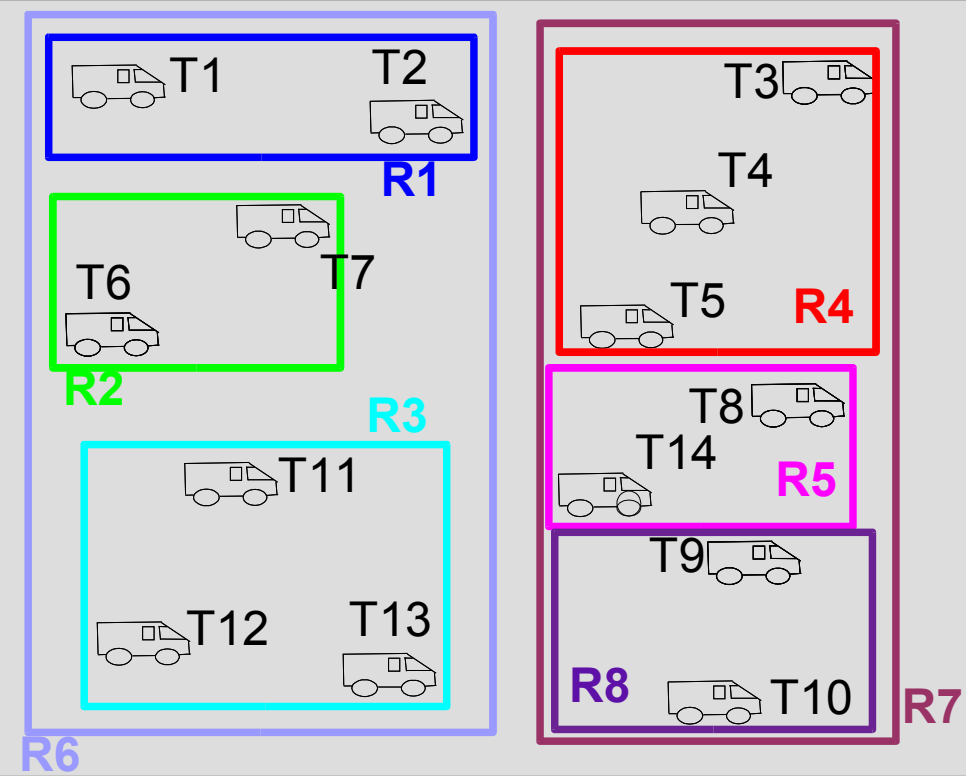


Steige auf solange es sich etwas ändert und passe den Knoten an.



Insert

Fertig !!!



T1 T2

T6 T7

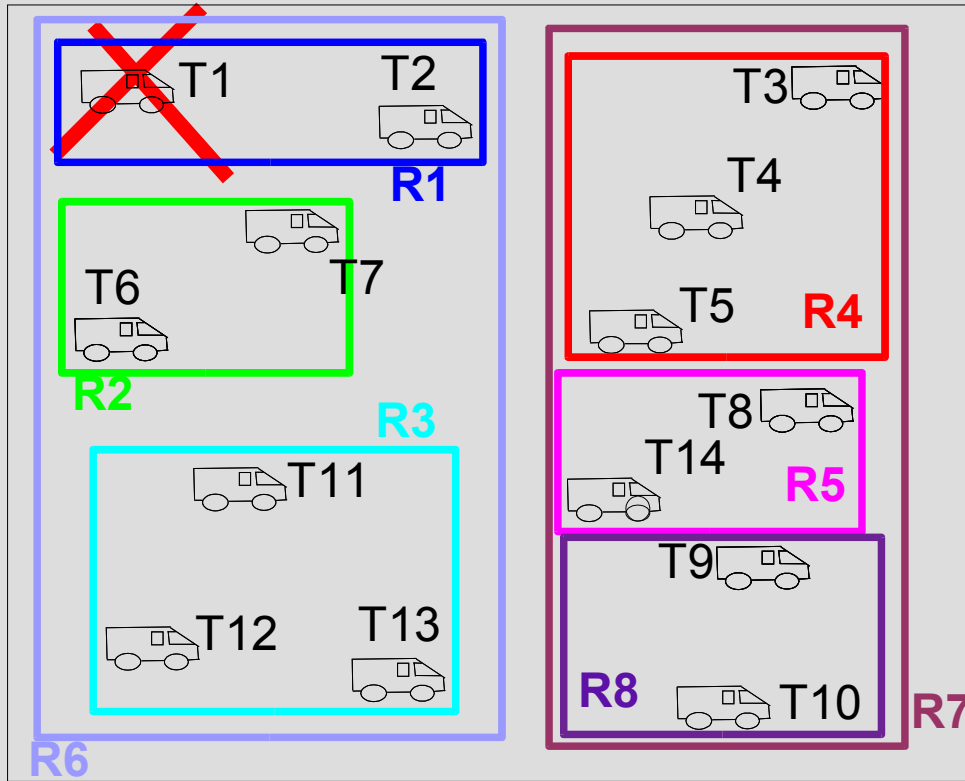
T11 T12 T13

T3 T4 T5

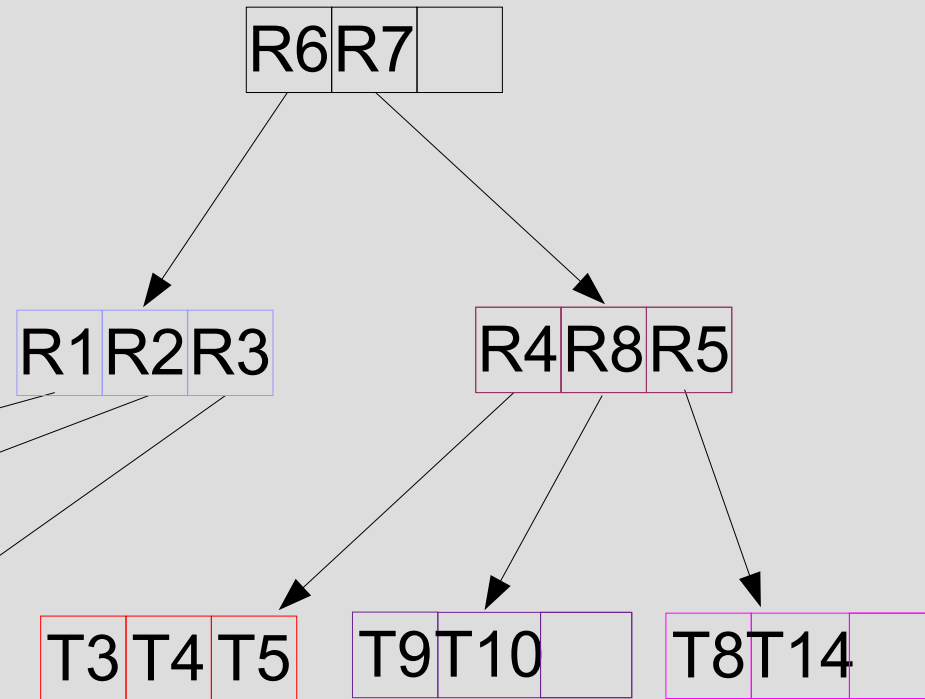
T9 T10

T8 T14

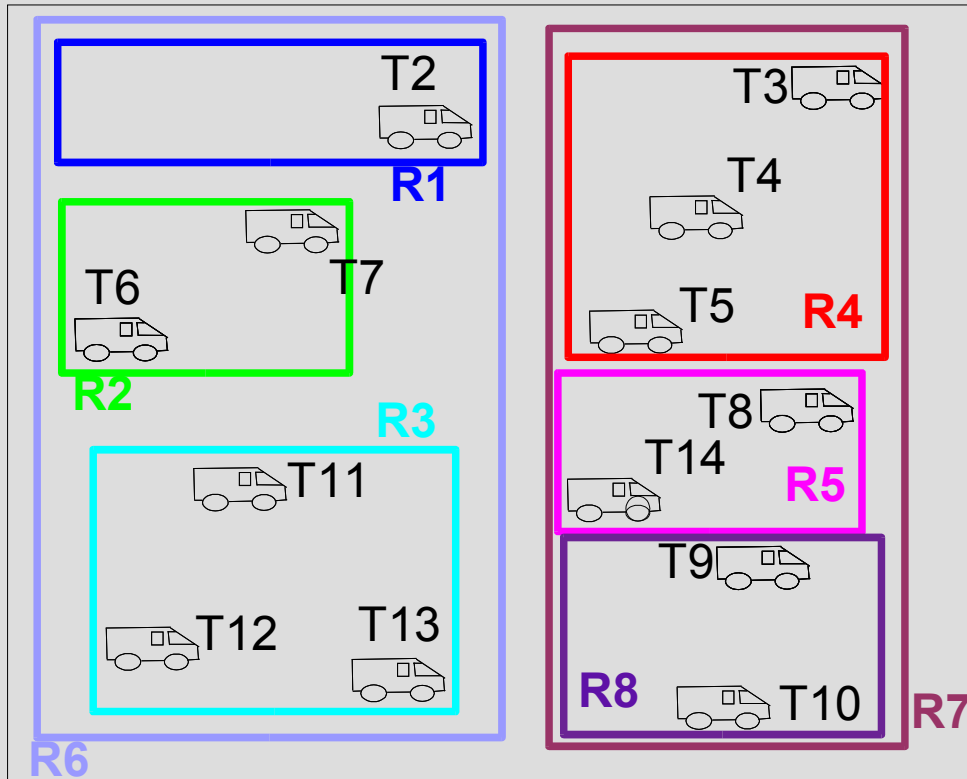
Delete



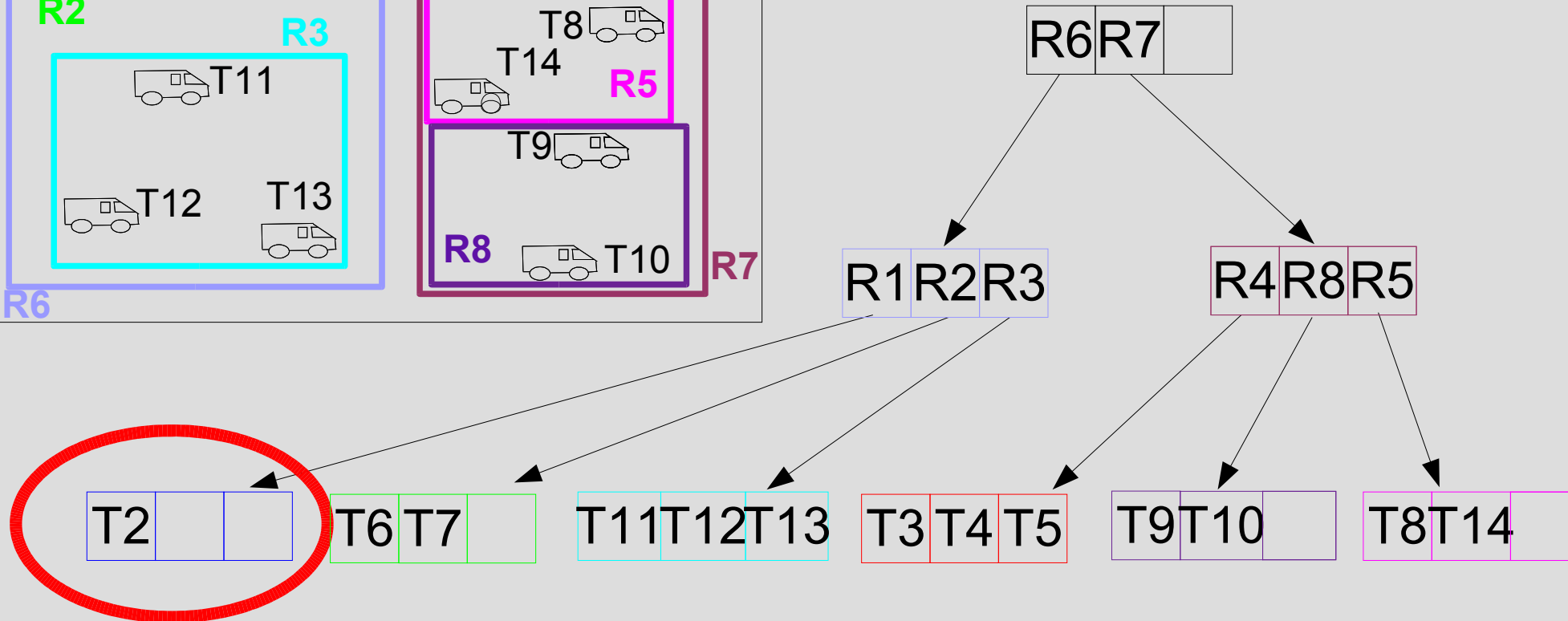
T1 fährt weg.



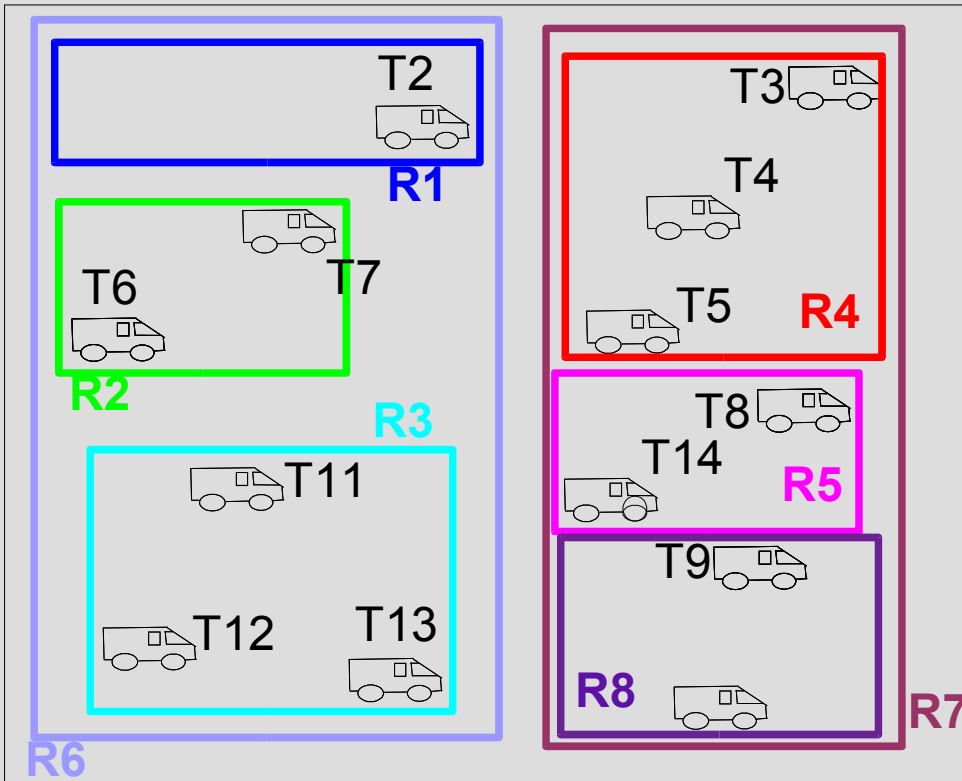
Delete



Knotenunterlauf!

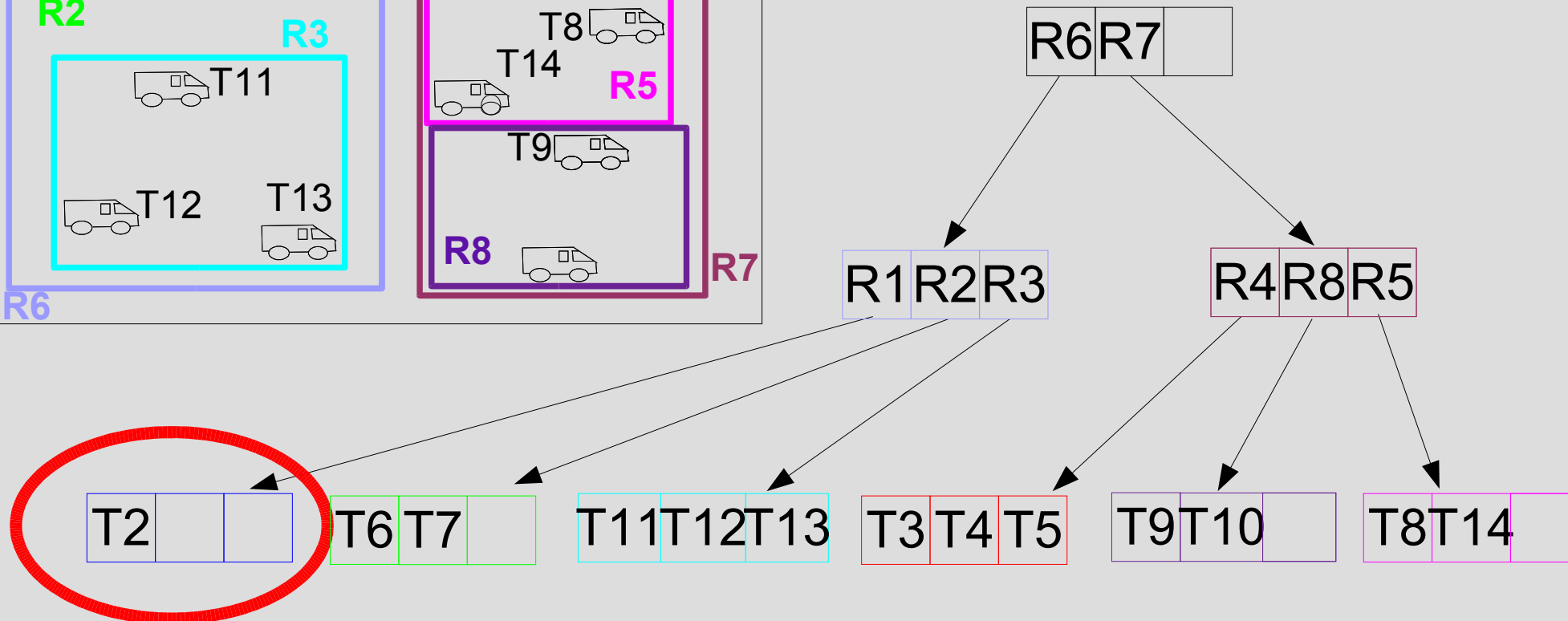


Delete

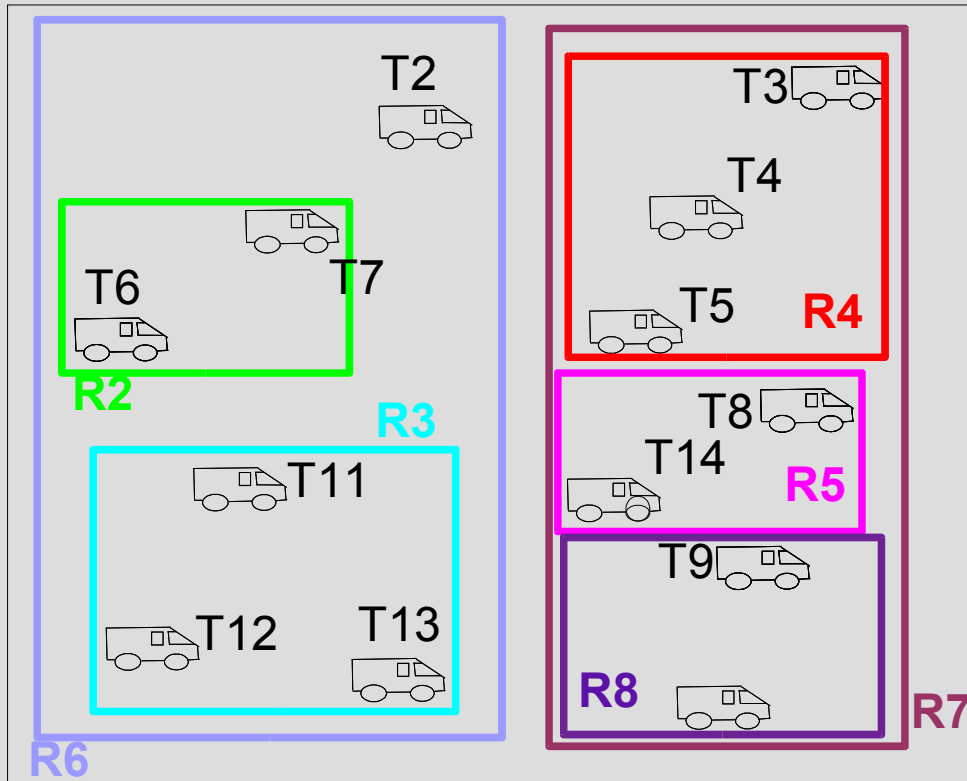


Knotenunterlauf!

=> Knoten anpassen

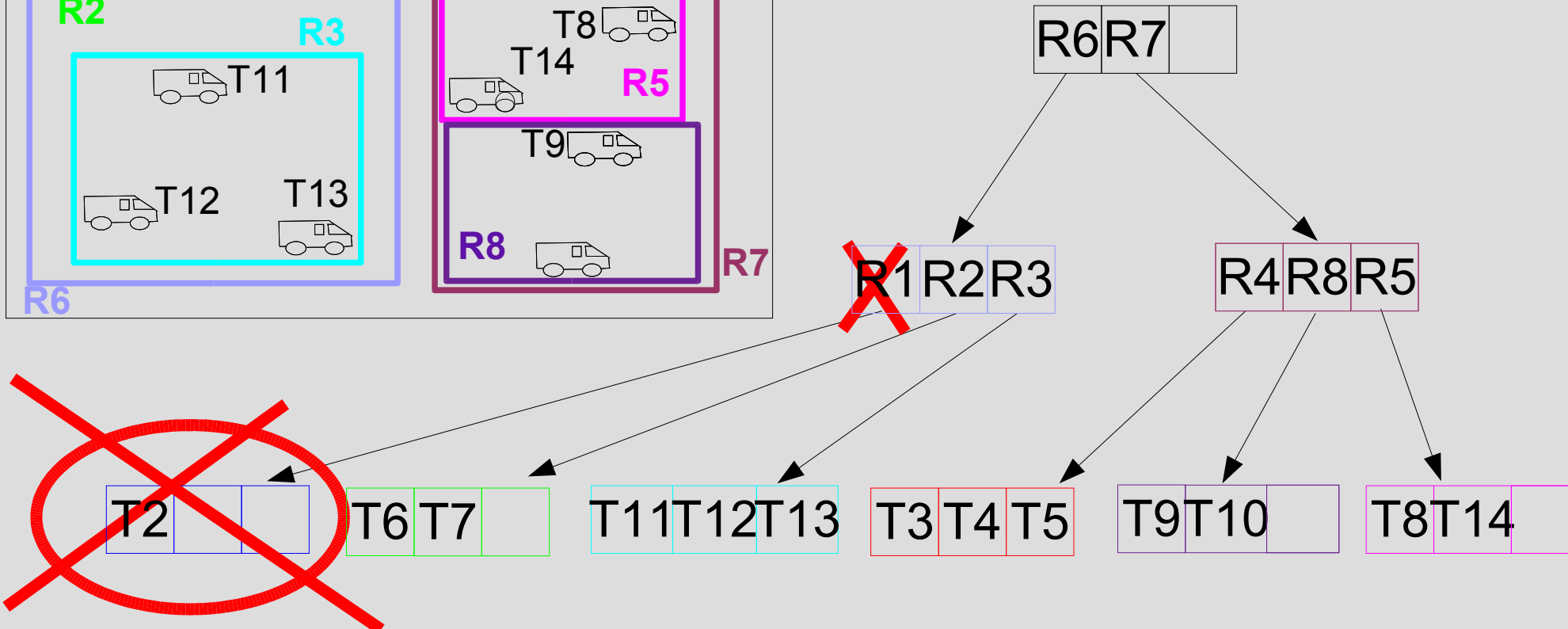


Delete

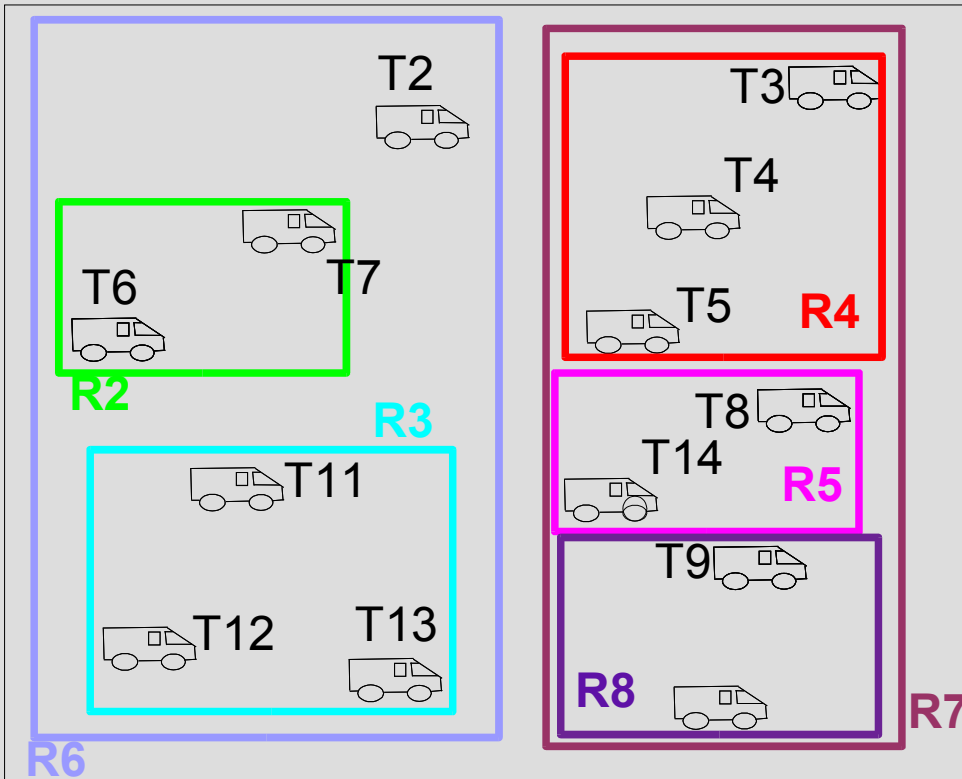


Knotenunterlauf!

=> Knoten löschen



Delete



Knotenunterlauf!

=> Knoten neu
einfügen (Insert)

T2



R6R7

R2R3

R4R8R5

T6 T7

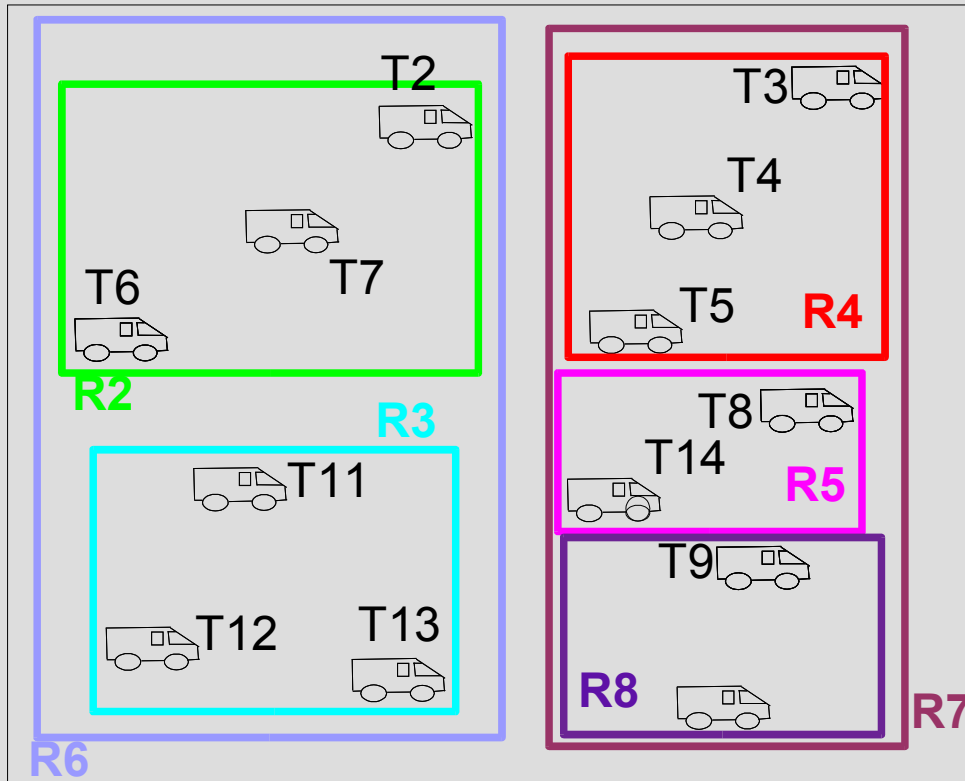
T11 T12 T13

T3 T4 T5

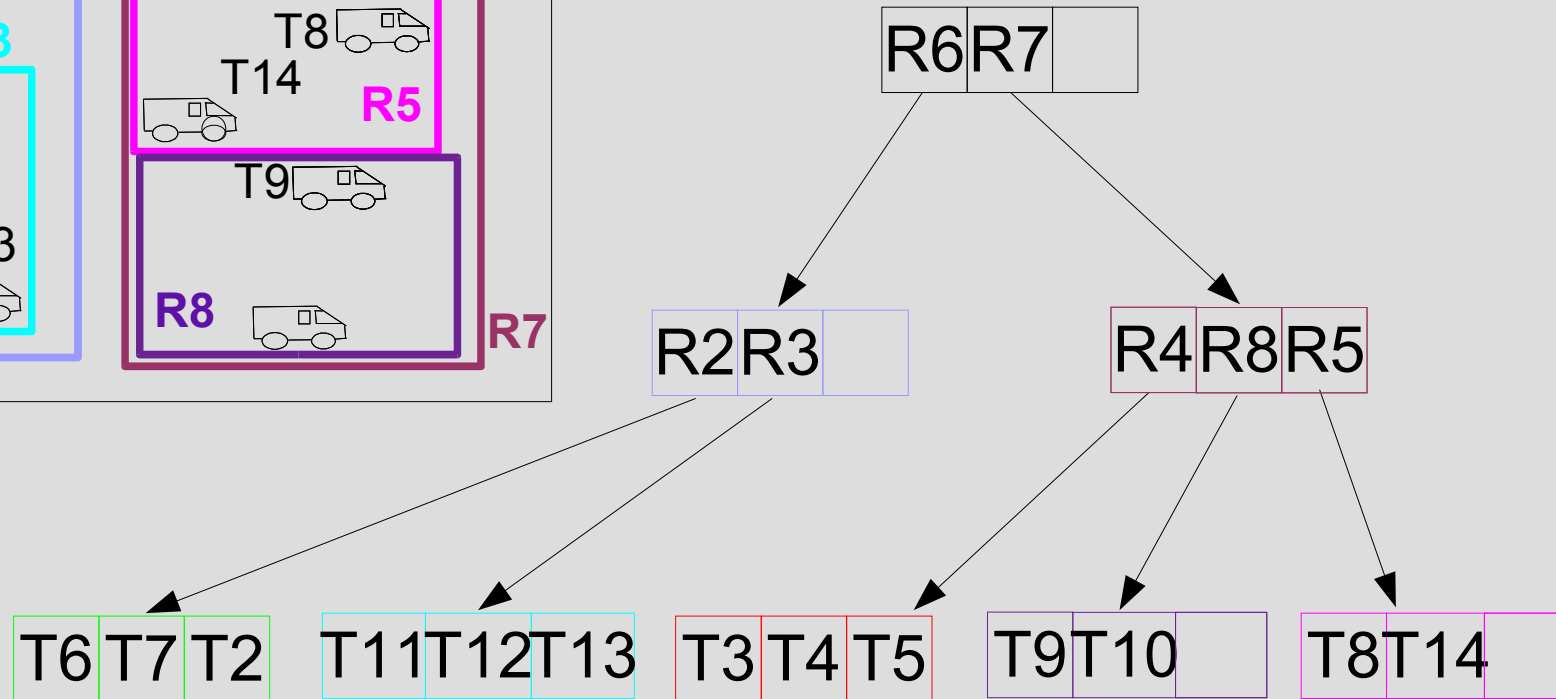
T9 T10

T8 T14

Delete

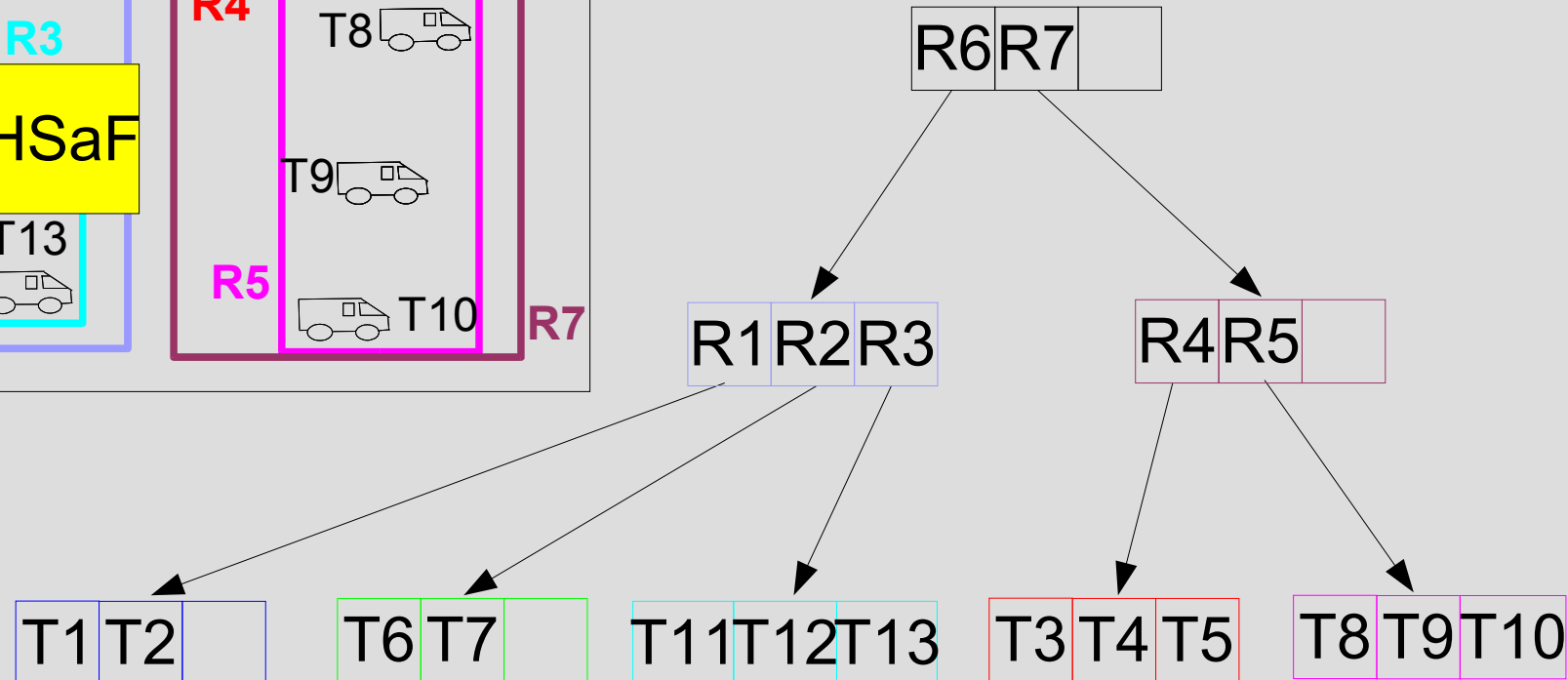
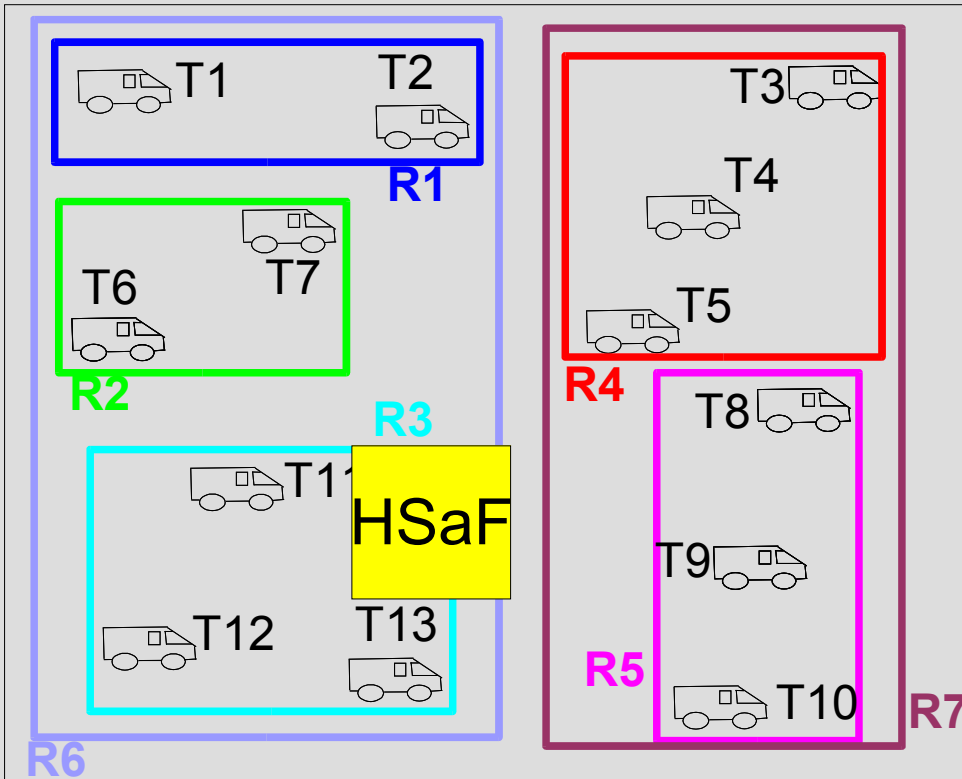


Fertig!



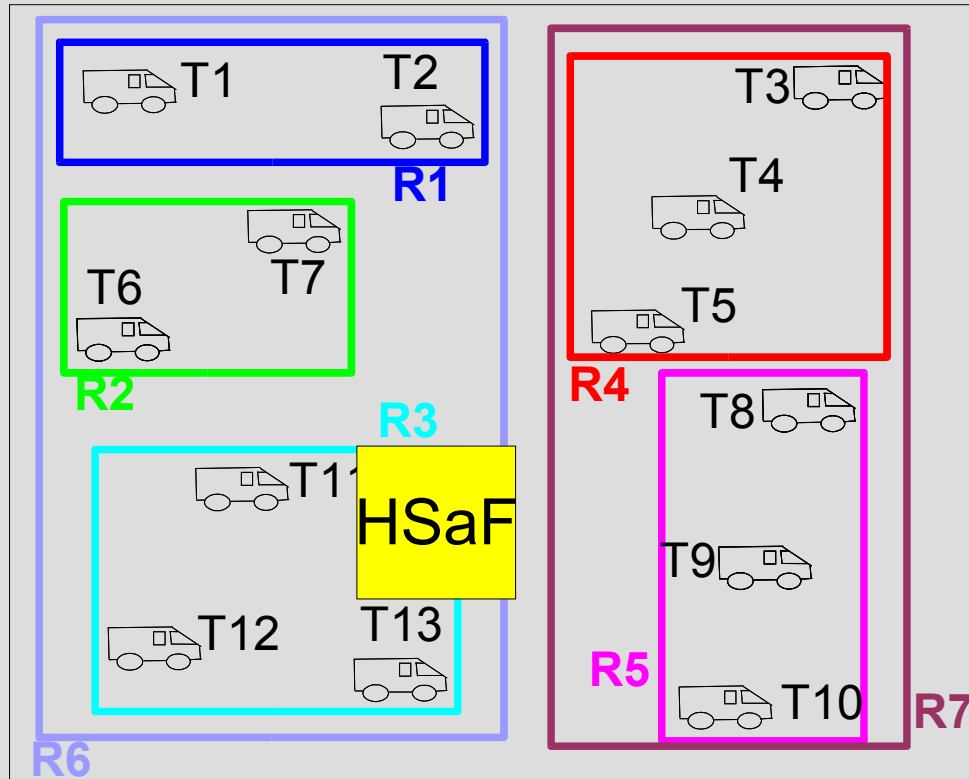
Anfrage

Welcher Transporter befindet sich in der Nähe des **HSaF**?

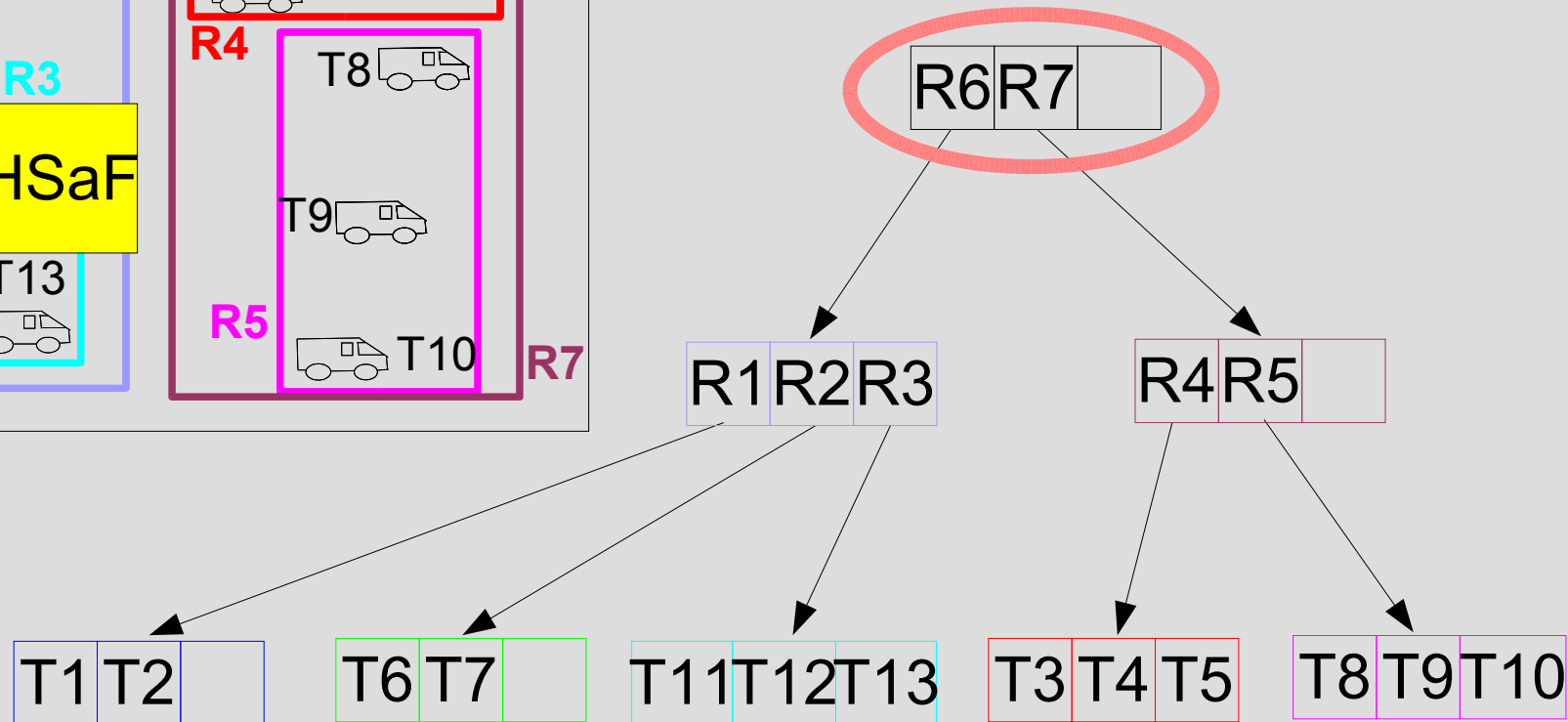


Anfrage

Welcher Transporter befindet sich in der Nähe des **HSaF**?

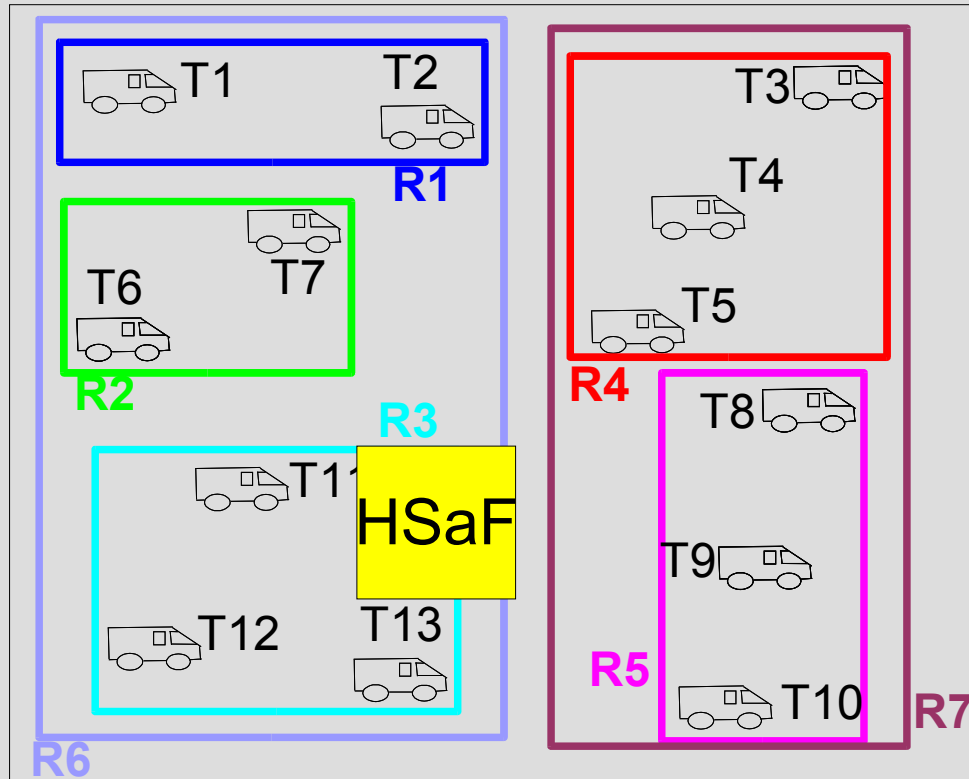


Überschneidung?



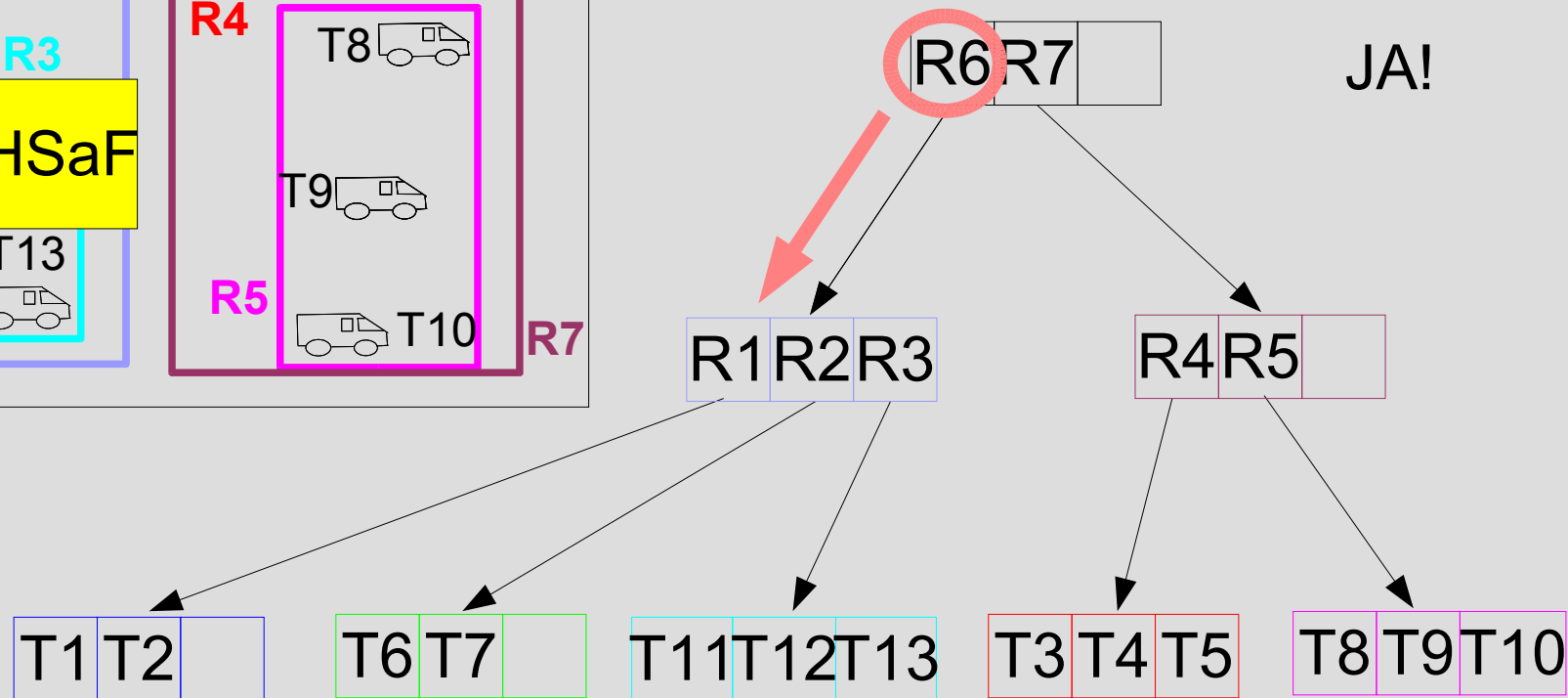
Anfrage

Welcher Transporter befindet sich in der Nähe des **HSaF**?



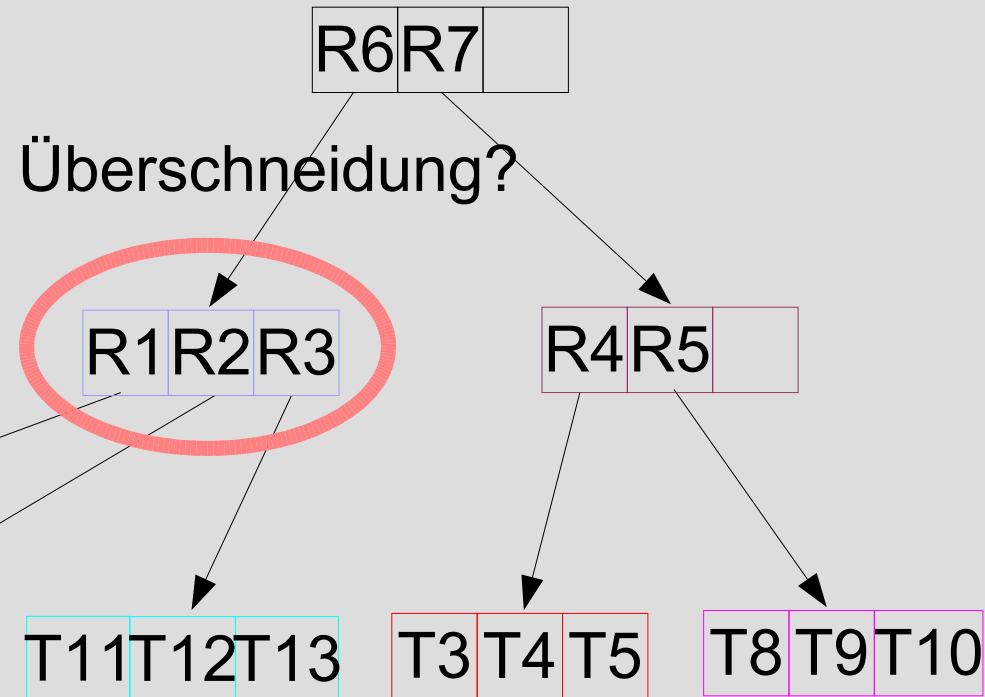
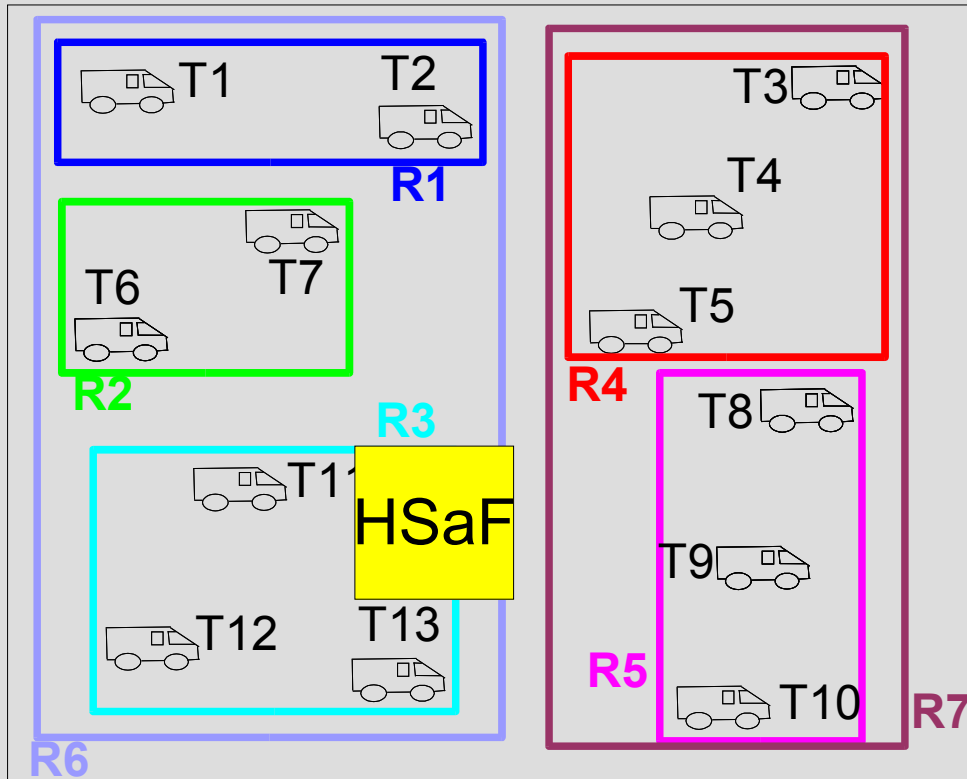
Überschneidung?

JA!



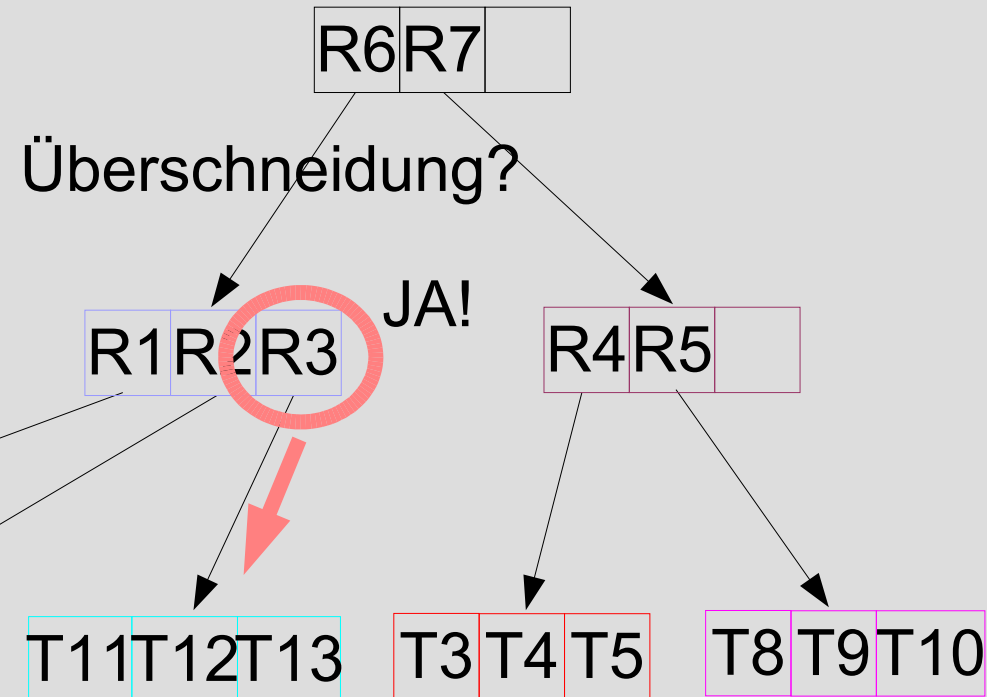
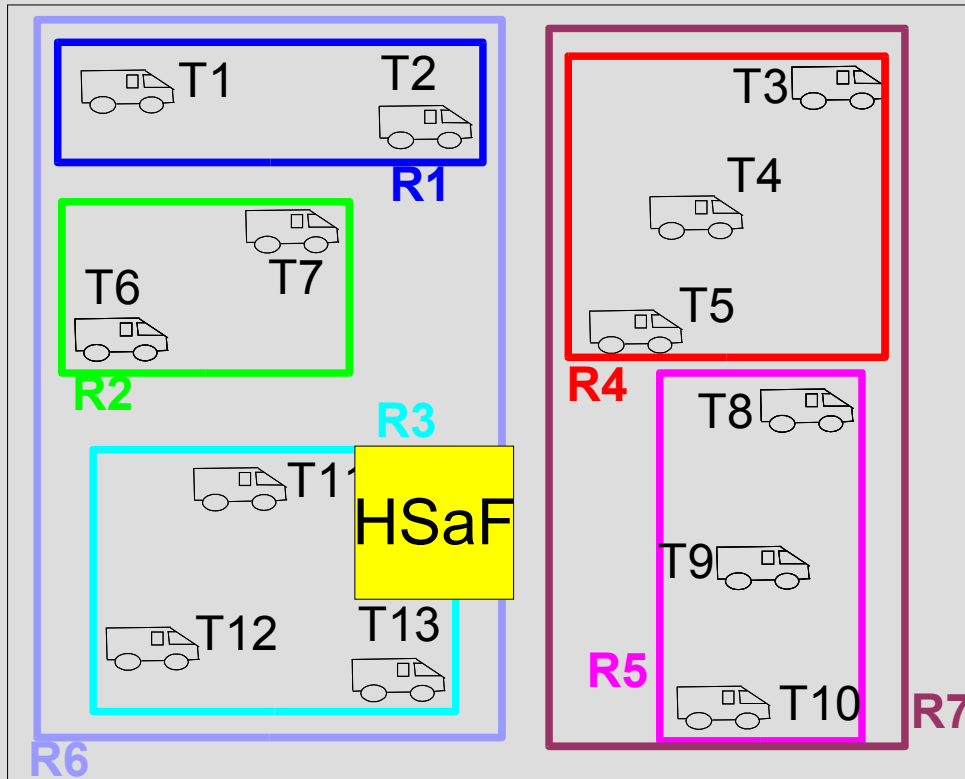
Anfrage

Welcher Transporter befindet sich in der Nähe des **HSaF**?



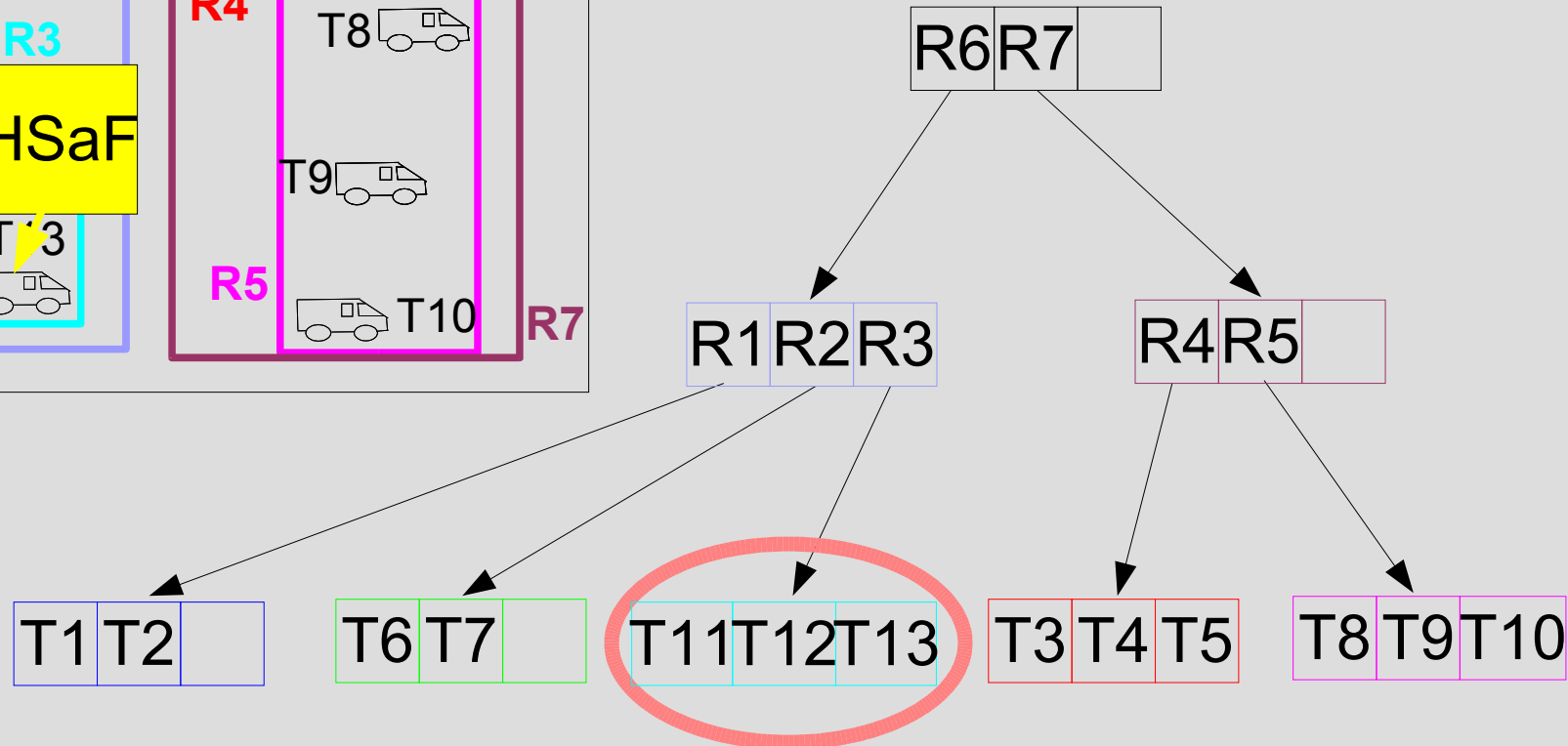
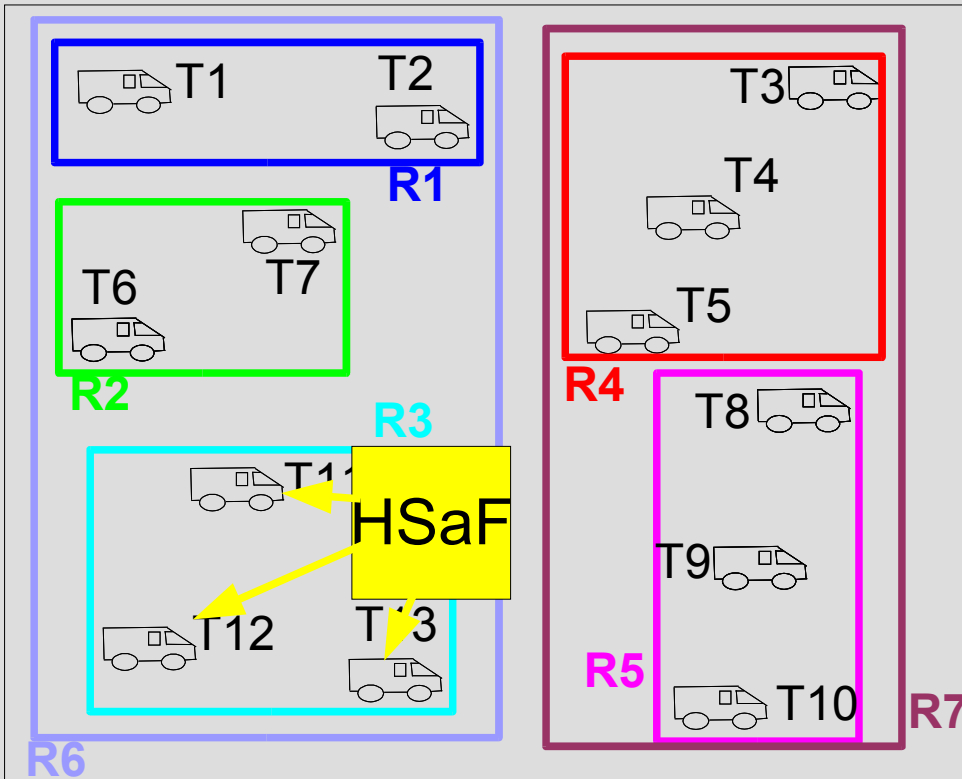
Anfrage

Welcher Transporter befindet sich in der Nähe des **HSaF**?



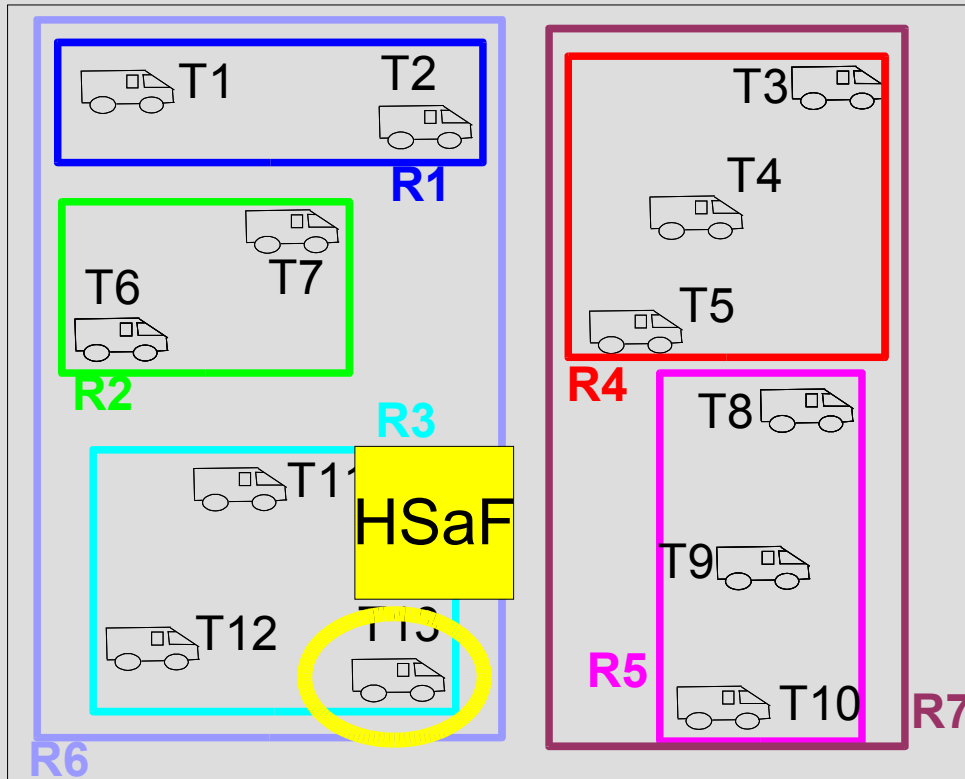
Anfrage

Welcher Transporter befindet sich in der Nähe des **HSaF**?

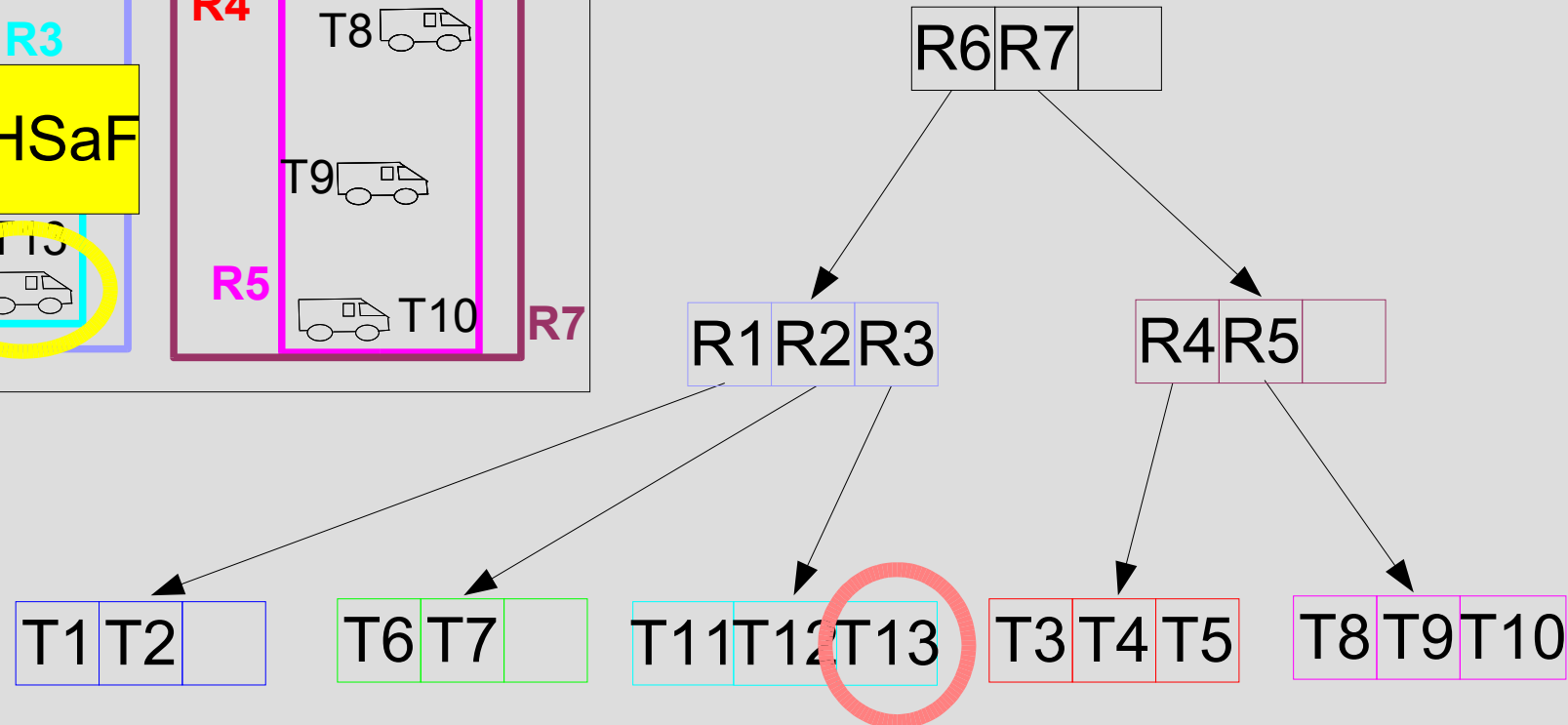


Anfrage

Welcher Transporter befindet sich in der Nähe des **HSaF**?



T13



Indexstrukturen

- B-Baum
 - keine effizienten räumlichen Suchanfragen möglich
- R-Baum
 - Erweiterung des B-Baum (mehrdimensionale Indexierung)
- R*-Baum
 - „Performance-optimierter“ R-Baum
- TPR-Baum
 - Anpassung des R*-Baum für „sich stetig fortbewegende Objekte“

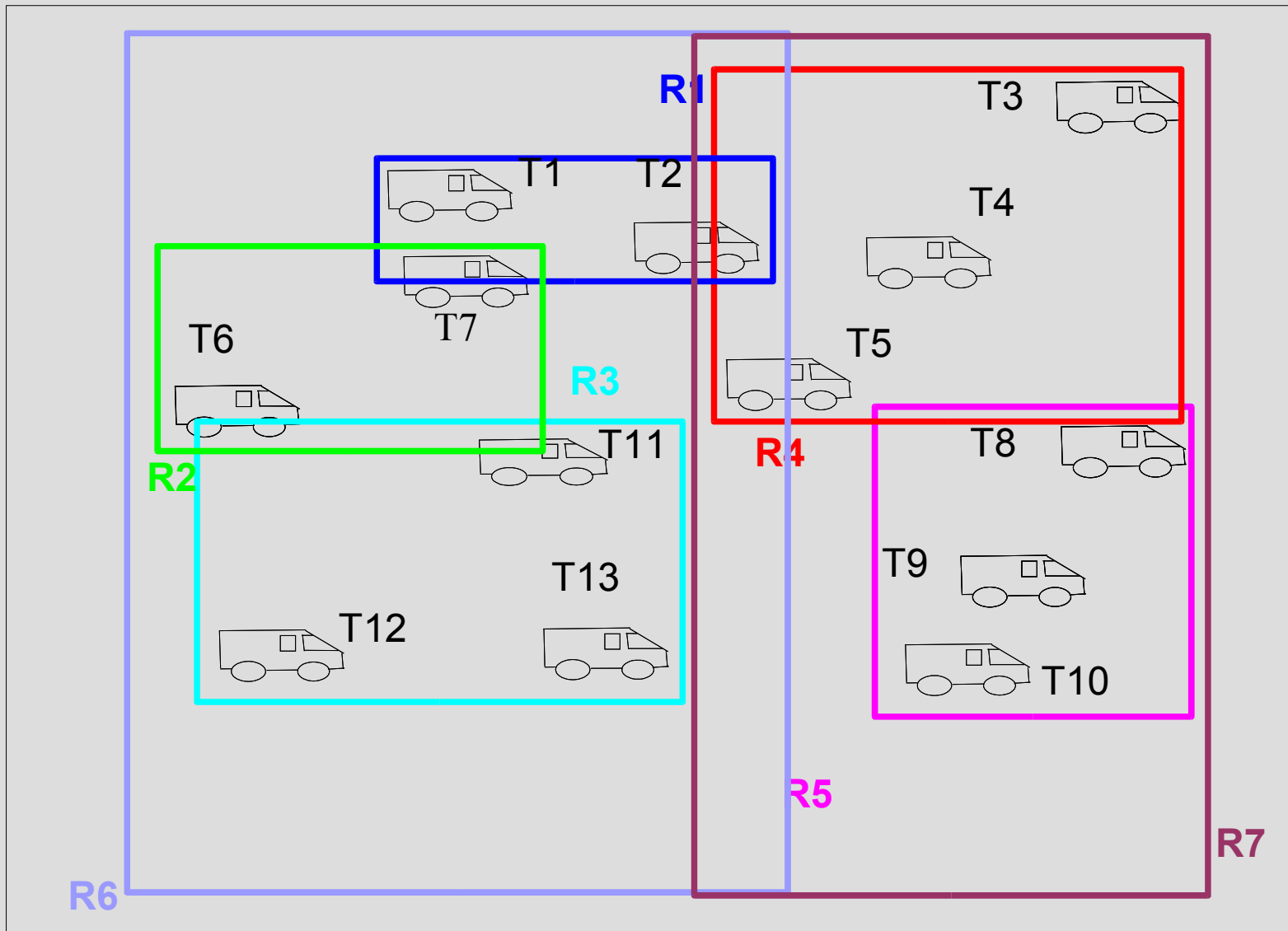
R*-Baum

- Verbesserung des R-Baum
- Unterschiede:
 - ▶ Ordnung
 - ▶ Insert
 - ChooseSubtree = ChooseLeaf
 - Reinsert
 - NodeSplit

Ordnung

- Vom R-Baum
 - ▶ **Minimiere Fläche** zwischen einem MBR und den darin enthaltenen Rechtecken
- Zusätzlich bei R^* -Baum
 - ▶ Minimiere die **Überlappungen** von Rechtecken
 - ▶ Minimiere den **Rand der Rechtecke**
 - ▶ Optimiere die **Speichernutzung**

Beispiel Überlappung



nach
mehrfachem
Einfügen
und
Löschen

ChooseLeaf

Insert:

ChooseLeaf;

if (Überlauf)

Reinsert;

if (Überlauf)

SplitNode;

1. Setze N gleich der Wurzel
2. If (N ist ein Blatt) return N
else if (die Zeiger in N zeigen auf Blätter)
wähle den Eintrag in N, dessen Rechteck die **wenigste Überlappungsvergrößerung** benötigt, um das neue Datenrechteck aufzunehmen.
else if (die Zeiger in N zeigen nicht auf Blätter)
wähle den Eintrag in N, dessen Rechteck die **wenigste Flächenvergrößerung** benötigt, um das neue Datenrechteck aufzunehmen.
3. Setze N auf den Zeiger des gewählten Eintrags zeigt und gehe zu 2.

Reinsert

Insert:

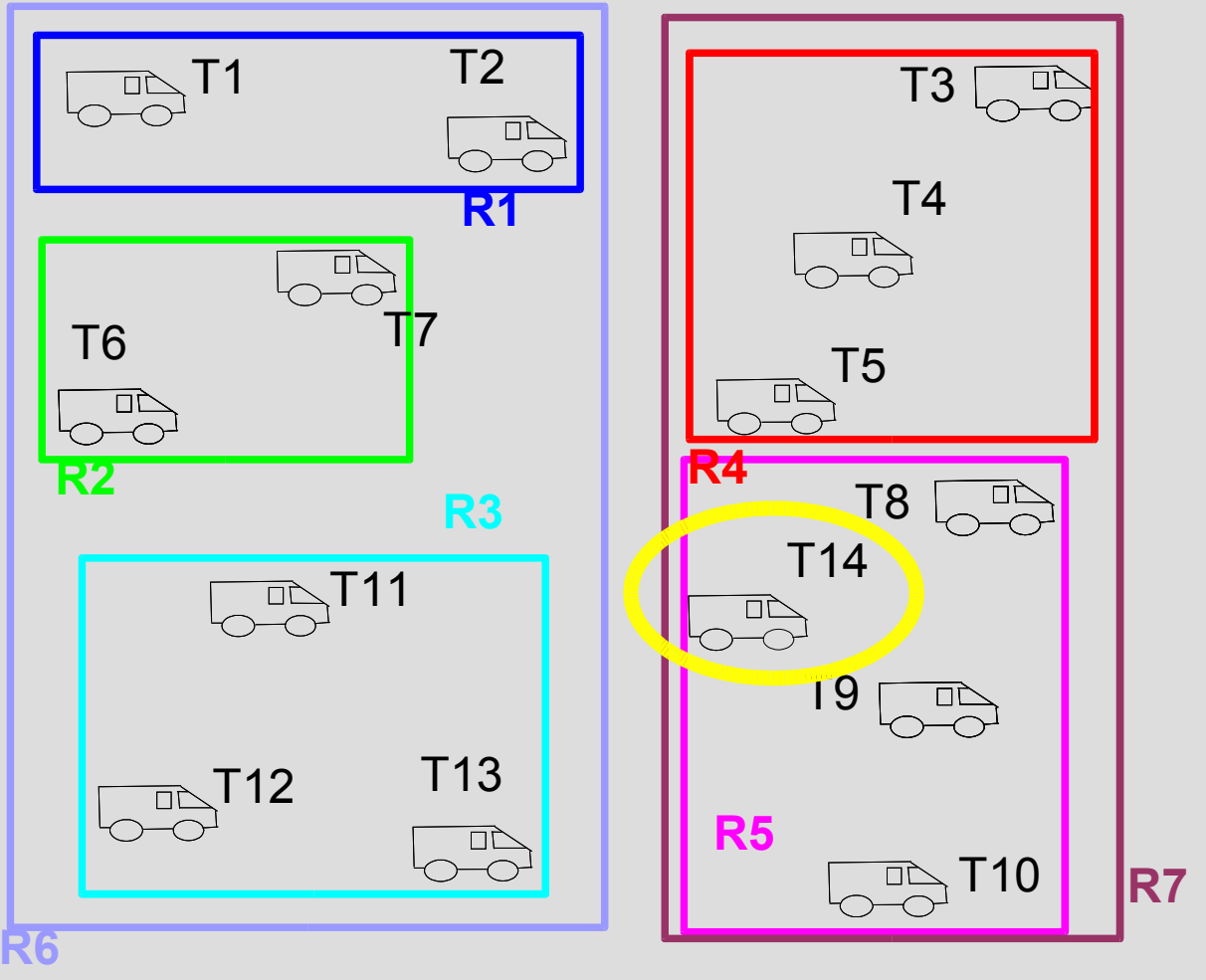
ChooseLeaf;

if (Überlauf)

Reinsert;

if (Überlauf)

NodeSplit;



Reinsert

Insert:

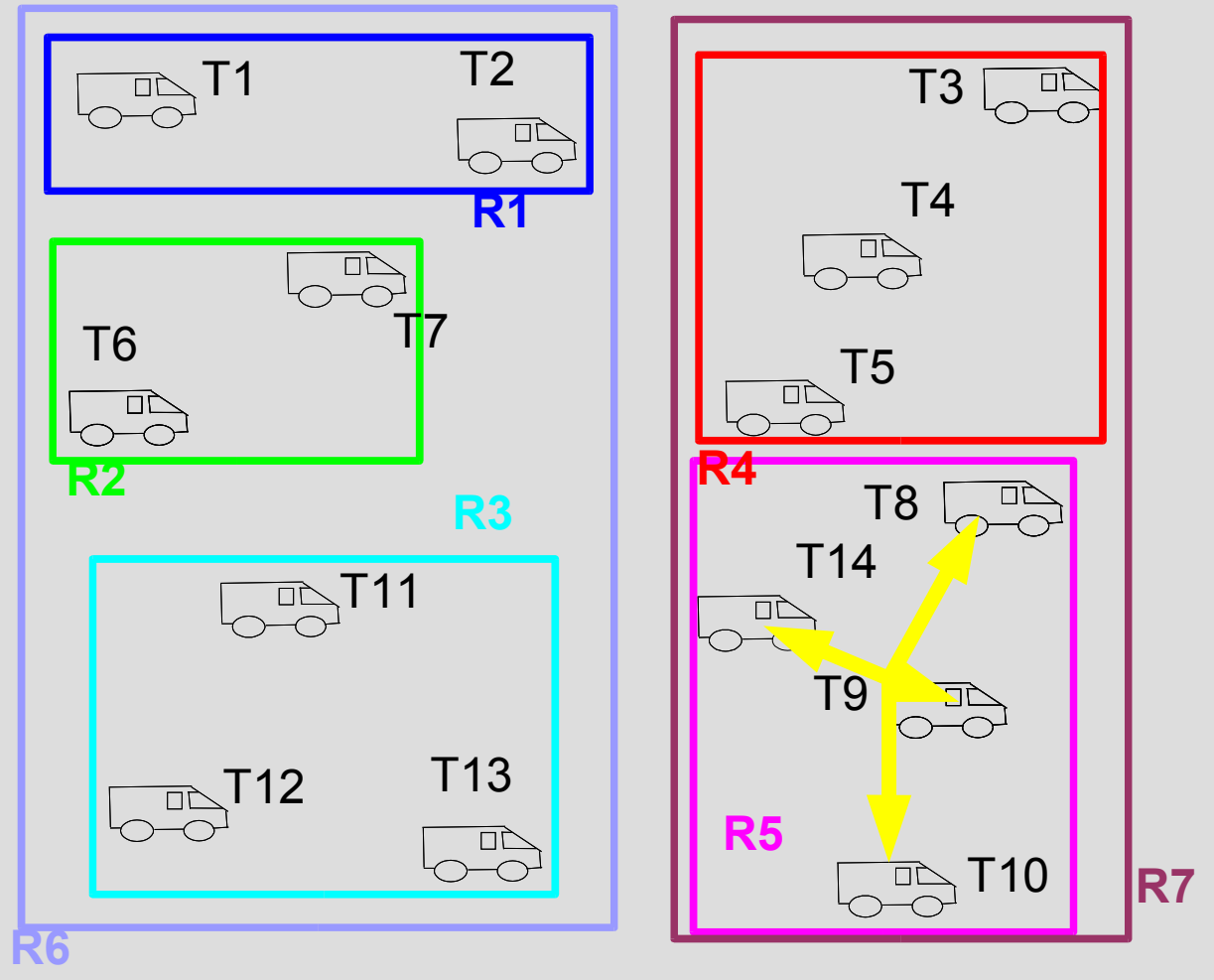
ChooseLeaf;

if (Überlauf)

Reinsert;

if (Überlauf)

NodeSplit;



Entfernung berechnen.

Reinsert

Insert:

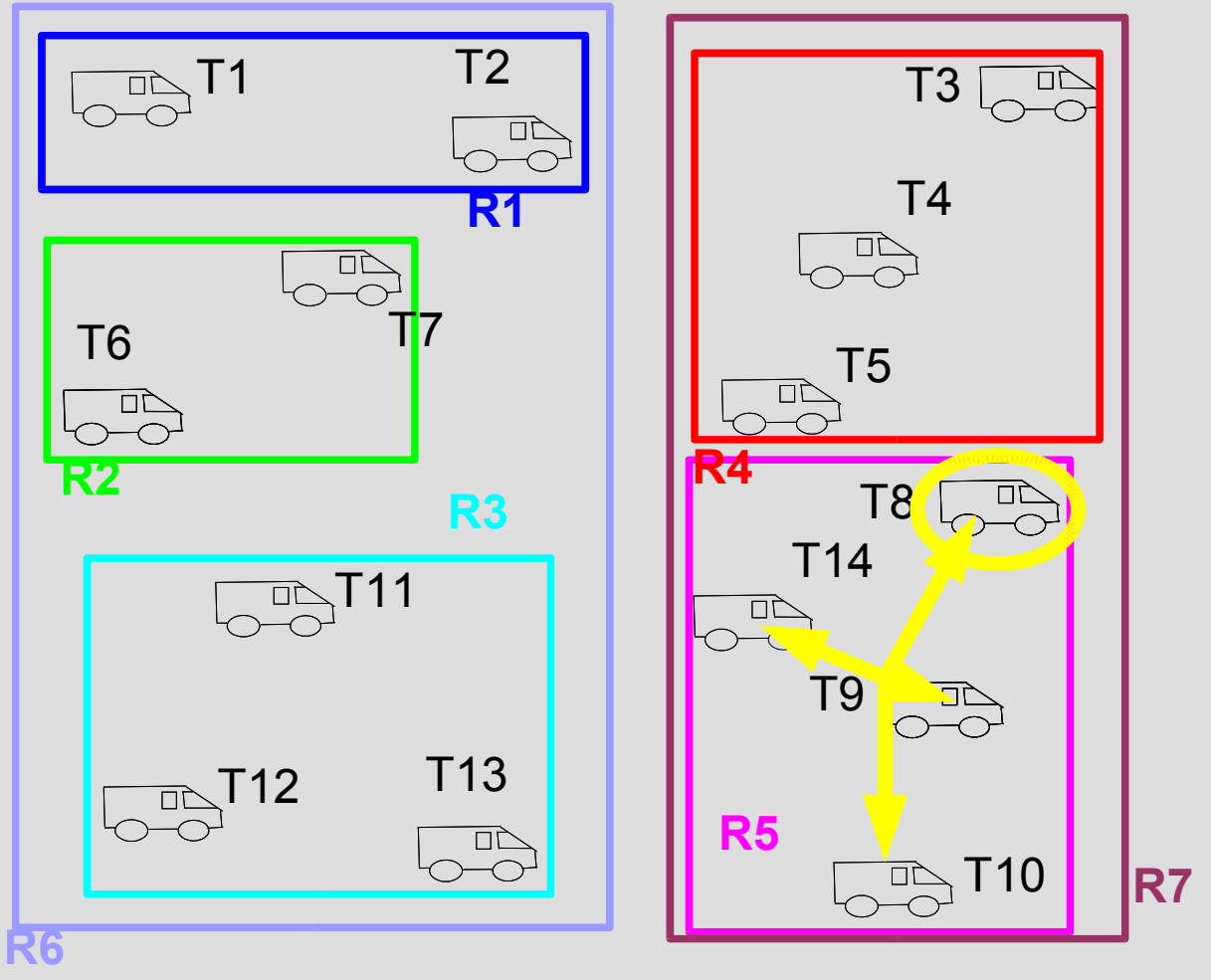
ChooseLeaf;

if (Überlauf)

Reinsert;

if (Überlauf)

NodeSplit;



Löschen.

Reinsert

Insert:

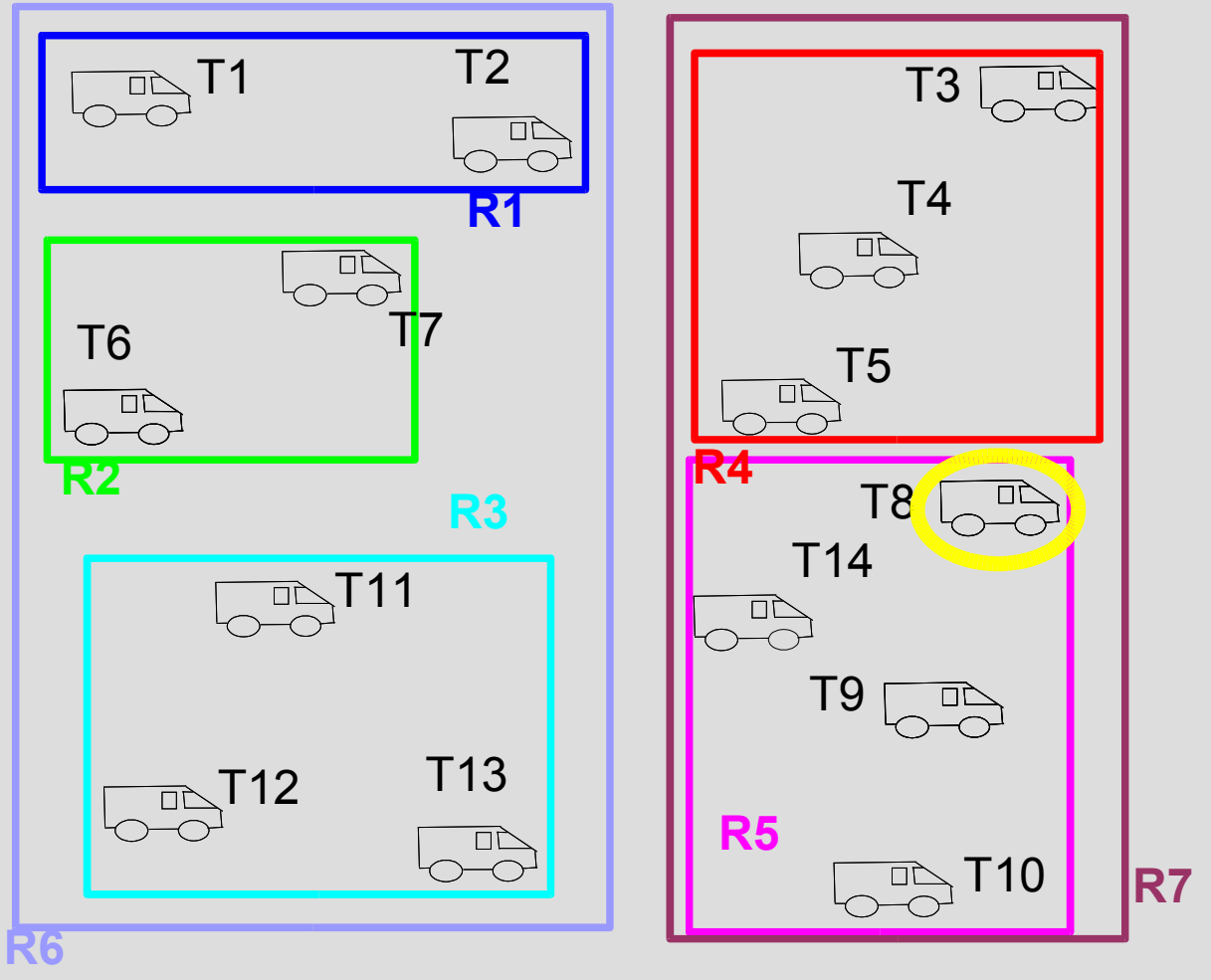
ChooseLeaf;

if (Überlauf)

Reinsert;

if (Überlauf)

NodeSplit;

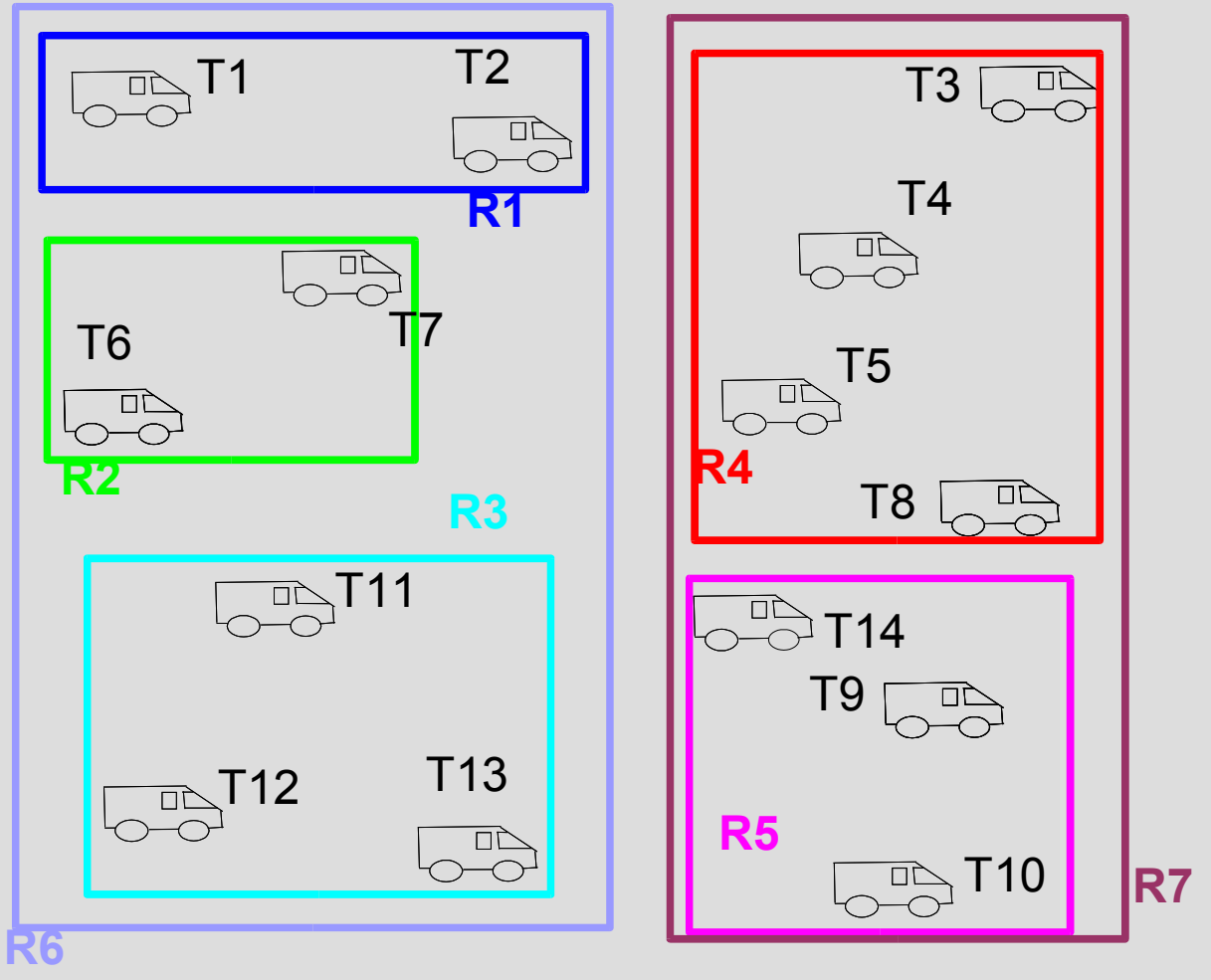


Wiedereinfügen.

Reinsert

Insert:

ChooseLeaf;
if (Überlauf)
Reinsert;
if (Überlauf)
NodeSplit;



Überlauf!!!

NodeSplit

Insert:

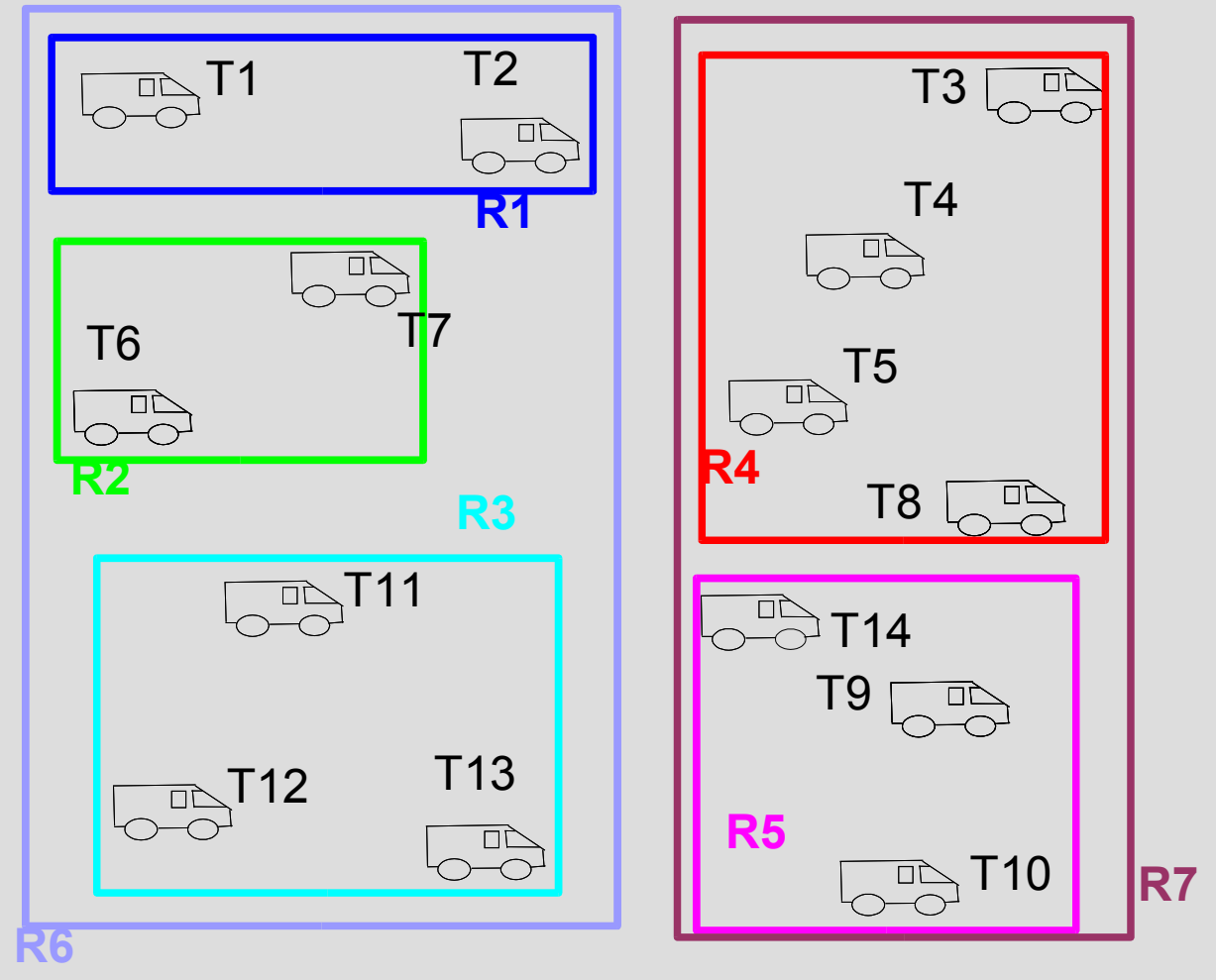
ChooseLeaf;

if (Überlauf)

Reinsert;

if (Überlauf)

NodeSplit;



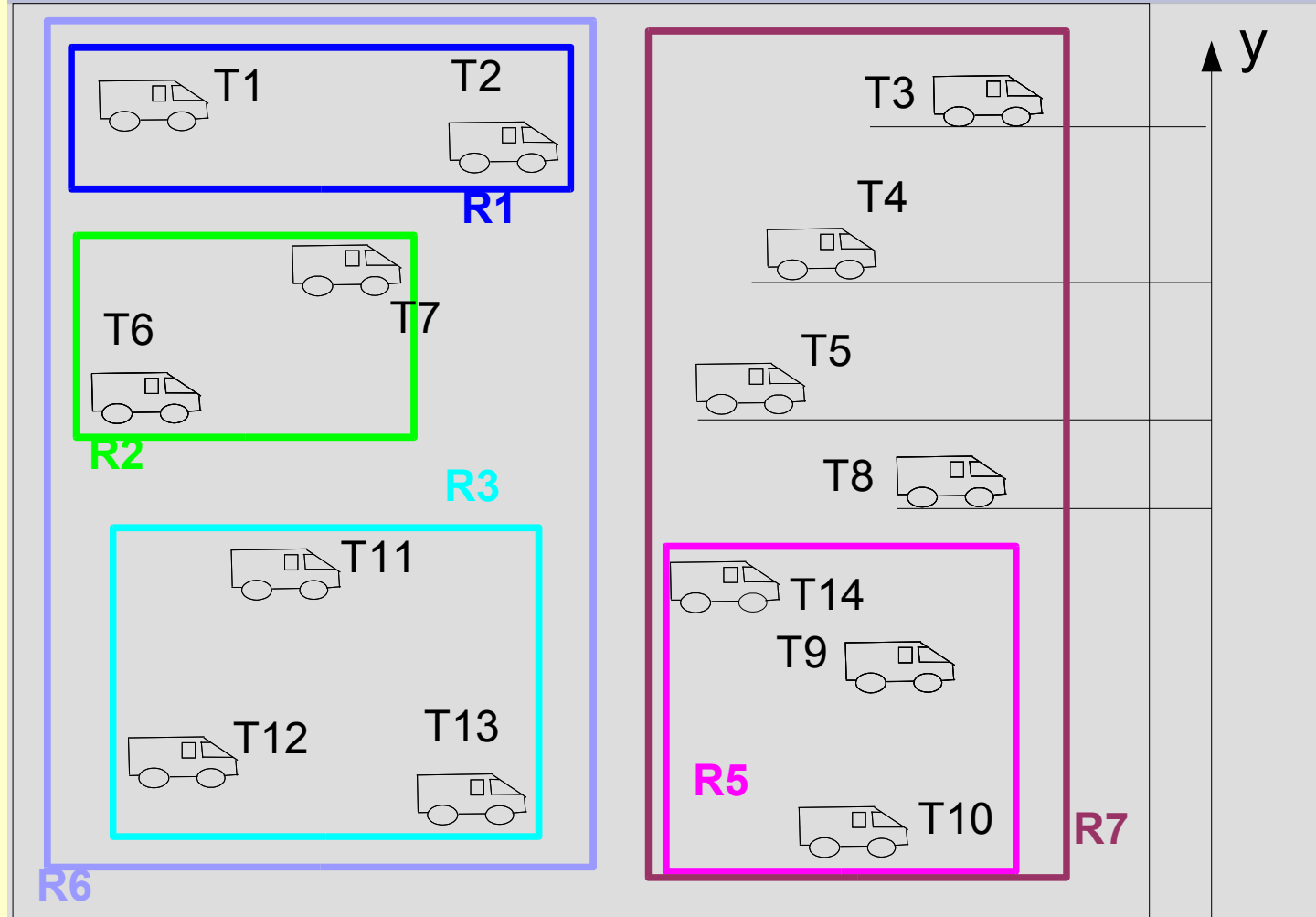
NodeSplit

NodeSplit:

ChooseAxis;

ChooseSplitIndex;

Teile auf;



Sortierung entlang der y-Achse (untere Werte).

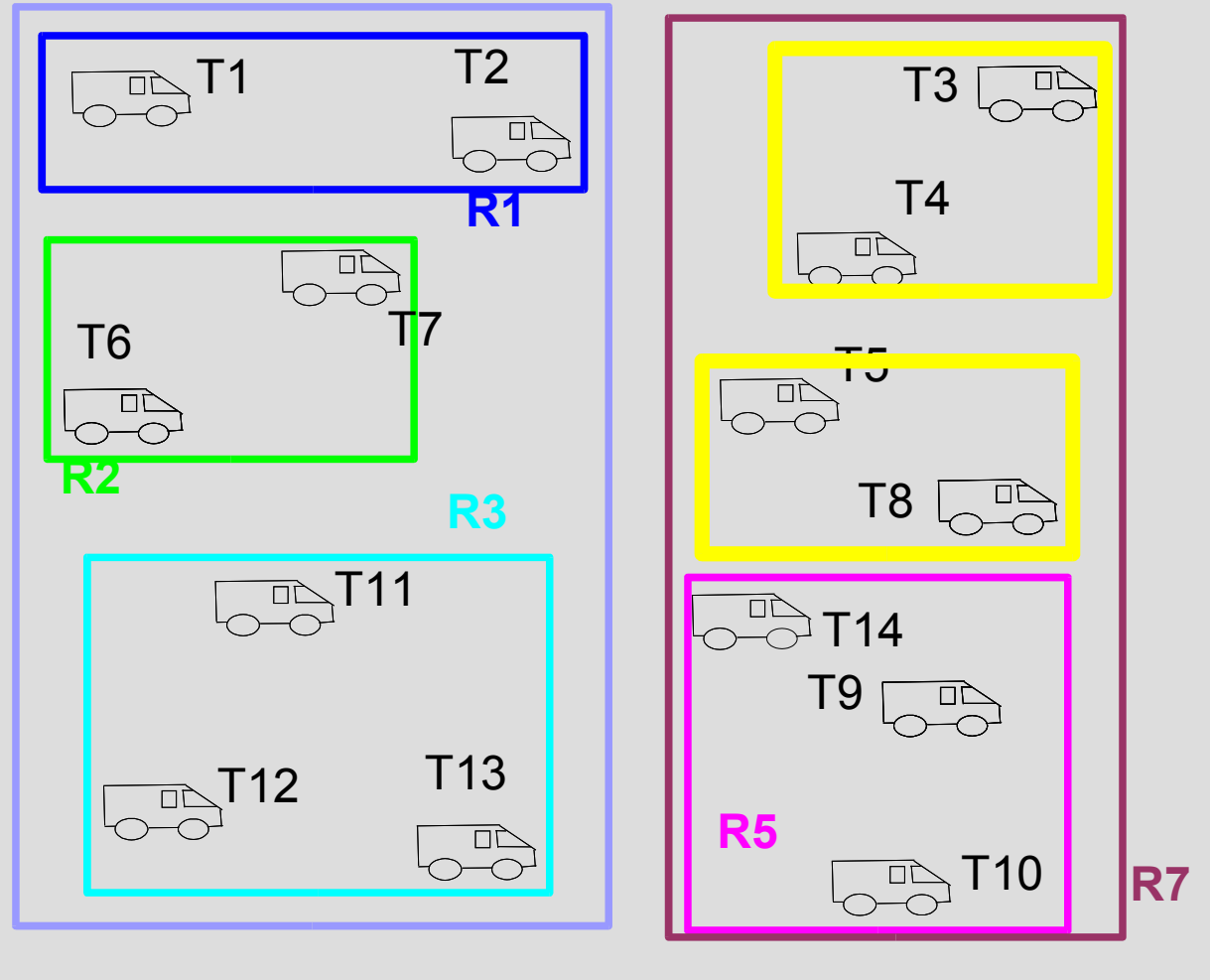
NodeSplit

NodeSplit:

ChooseAxis;

ChooseSplitIndex;

Teile auf;



Verteilungen für Sortierung untere Werte:
Es gibt nur eine!

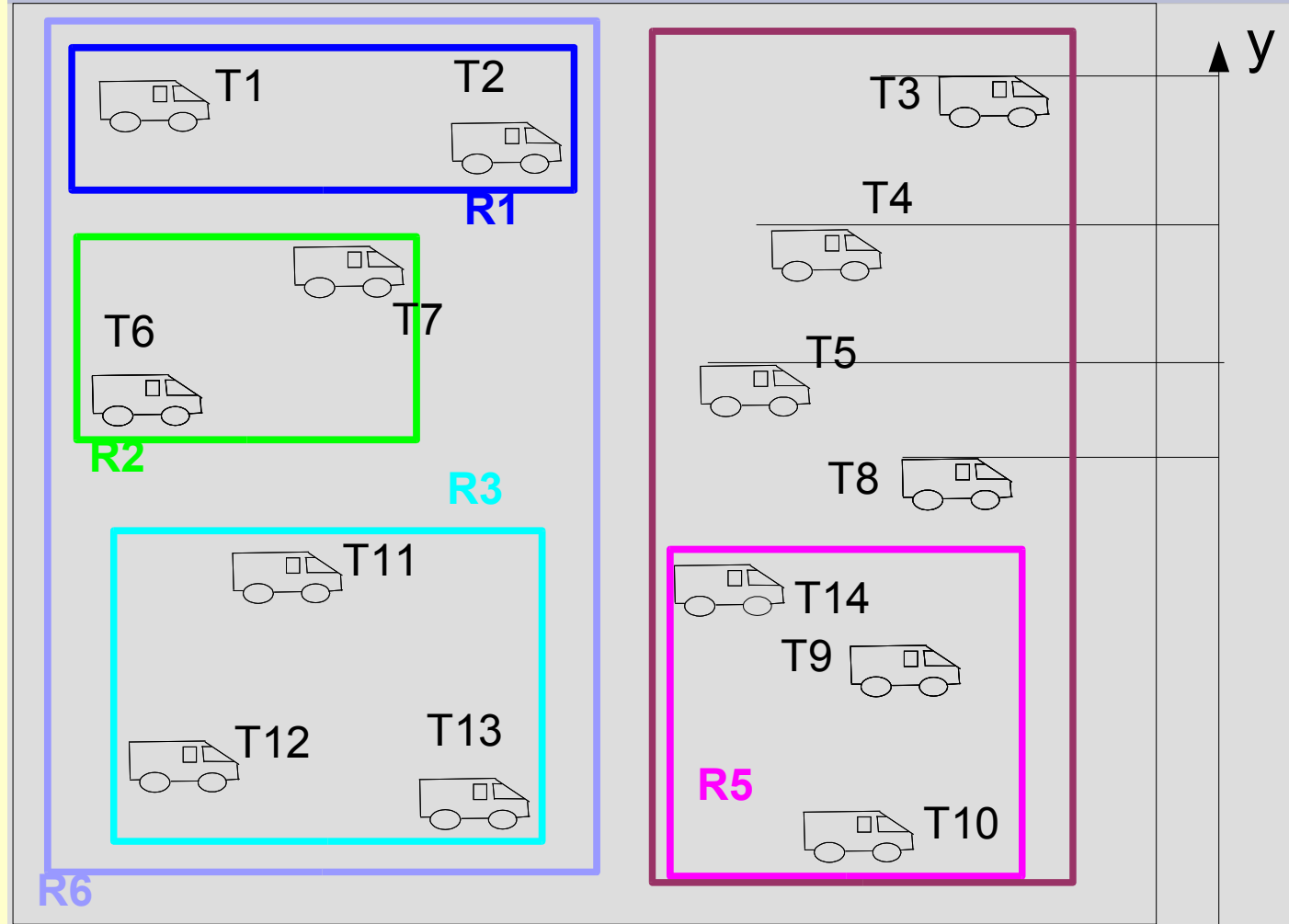
NodeSplit

NodeSplit:

ChooseAxis;

ChooseSplitIndex;

Teile auf;



Sortierung entlang der y-Achse (obere Werte).

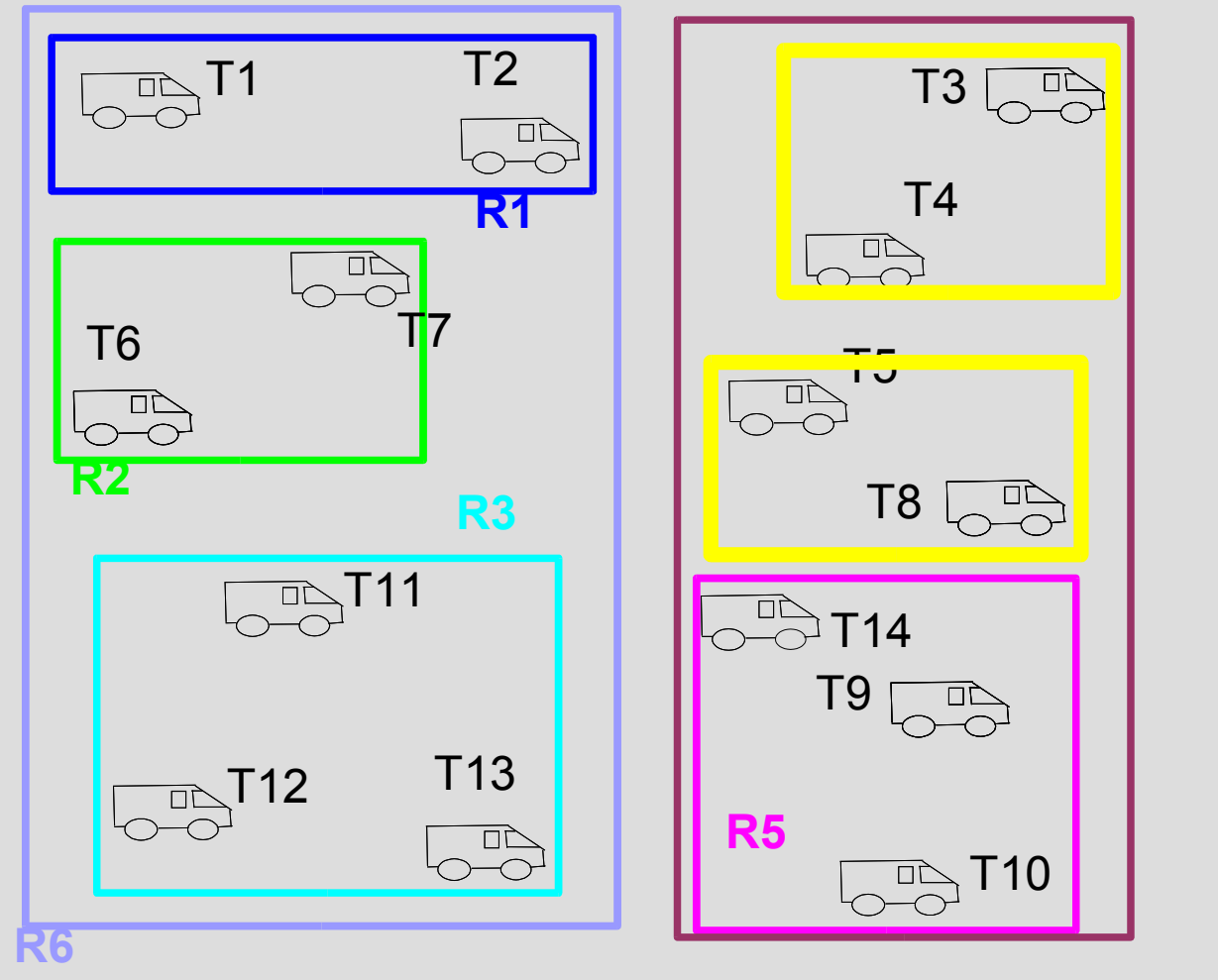
NodeSplit

NodeSplit:

ChooseAxis;

ChooseSplitIndex;

Teile auf;



Verteilungen für Sortierung oberer Werte:
Es gibt nur eine!

NodeSplit

NodeSplit:

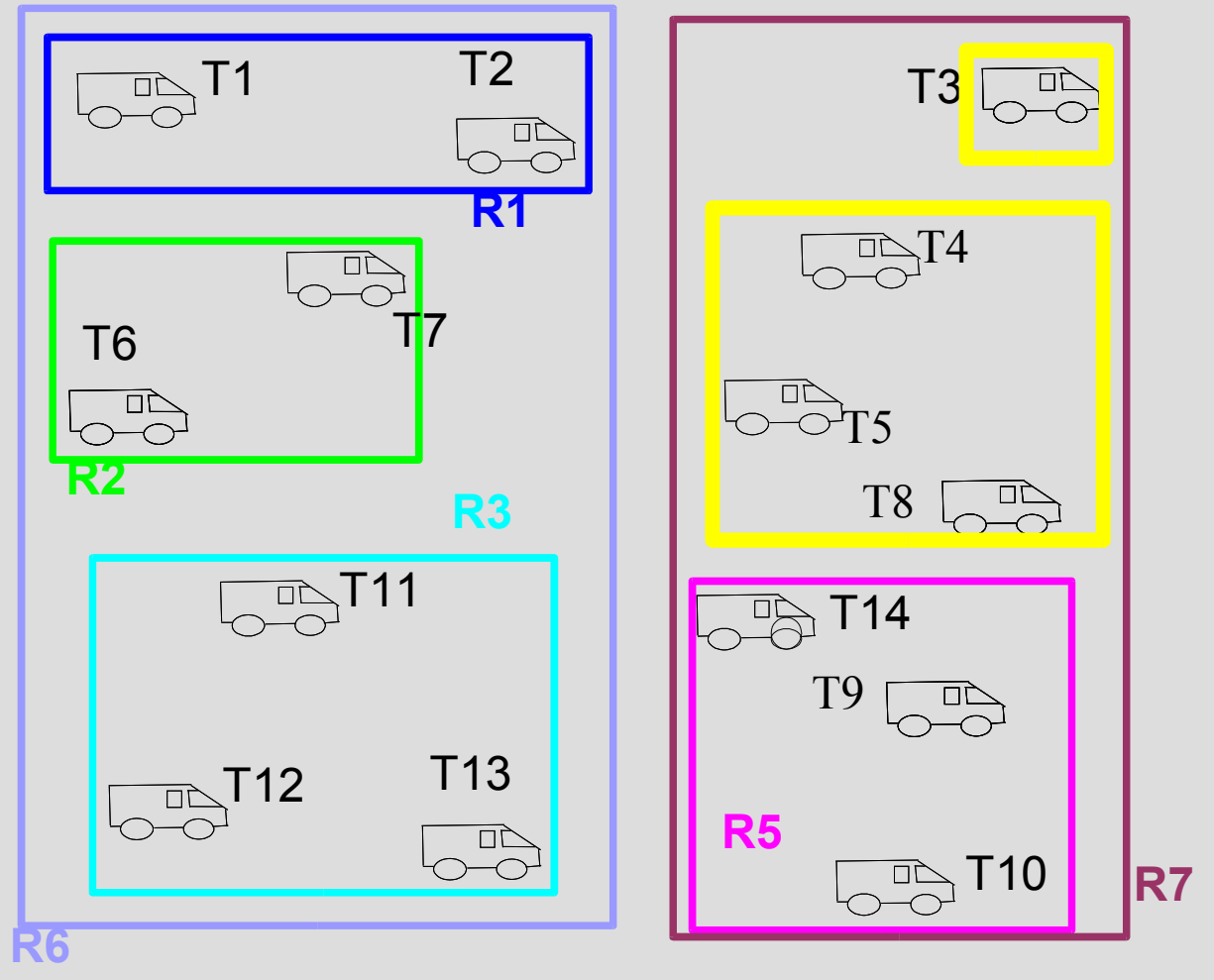
ChooseAxis;

ChooseSplitIndex;

Teile auf;

Annahme:

$1 \leq \text{Füllgrad} \leq 3$



NodeSplit

NodeSplit:

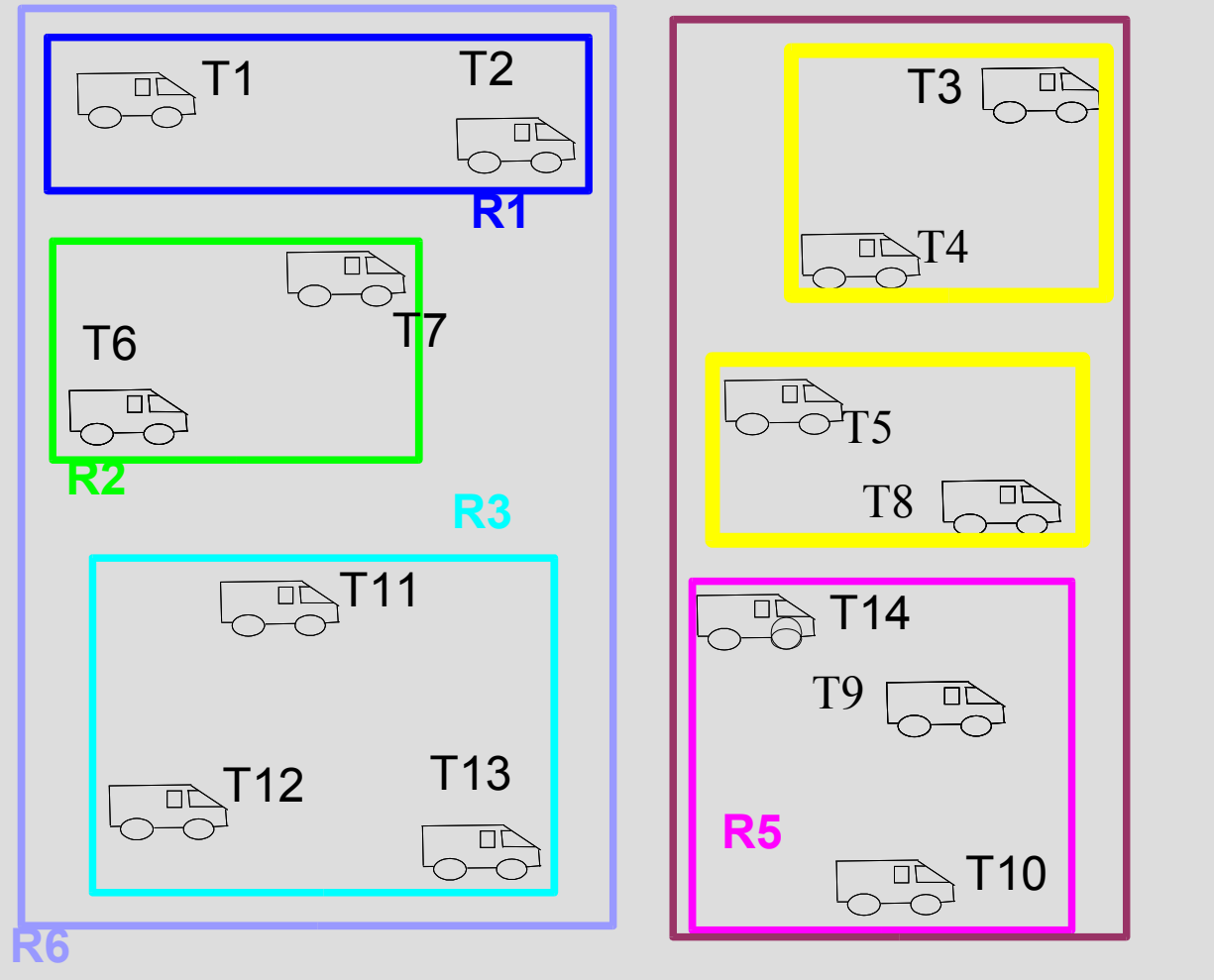
ChooseAxis;

ChooseSplitIndex;

Teile auf;

Annahme:

$1 \leq \text{Füllgrad} \leq 3$



NodeSplit

NodeSplit:

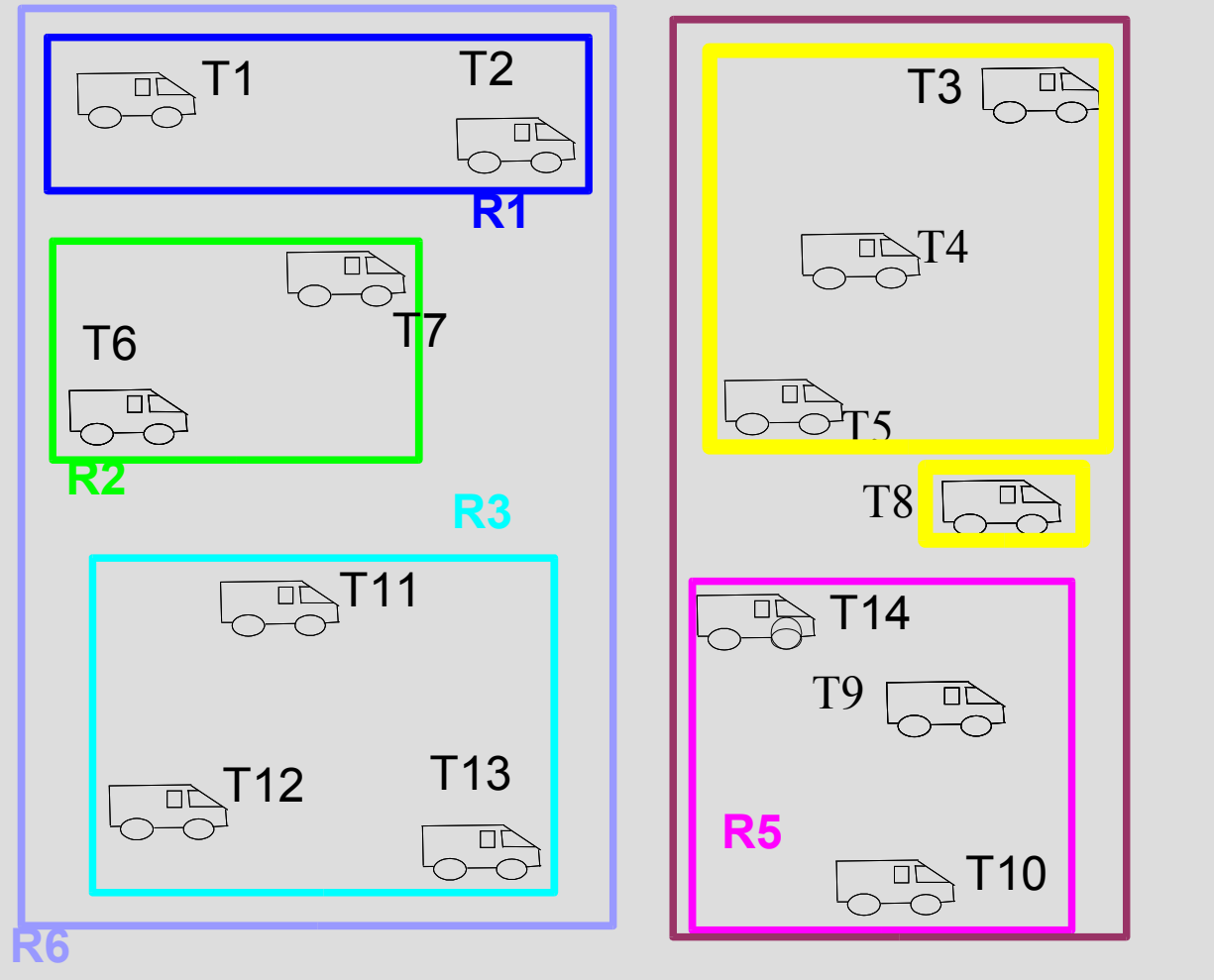
ChooseAxis;

ChooseSplitIndex;

Teile auf;

Annahme:

$1 \leq \text{Füllgrad} \leq 3$



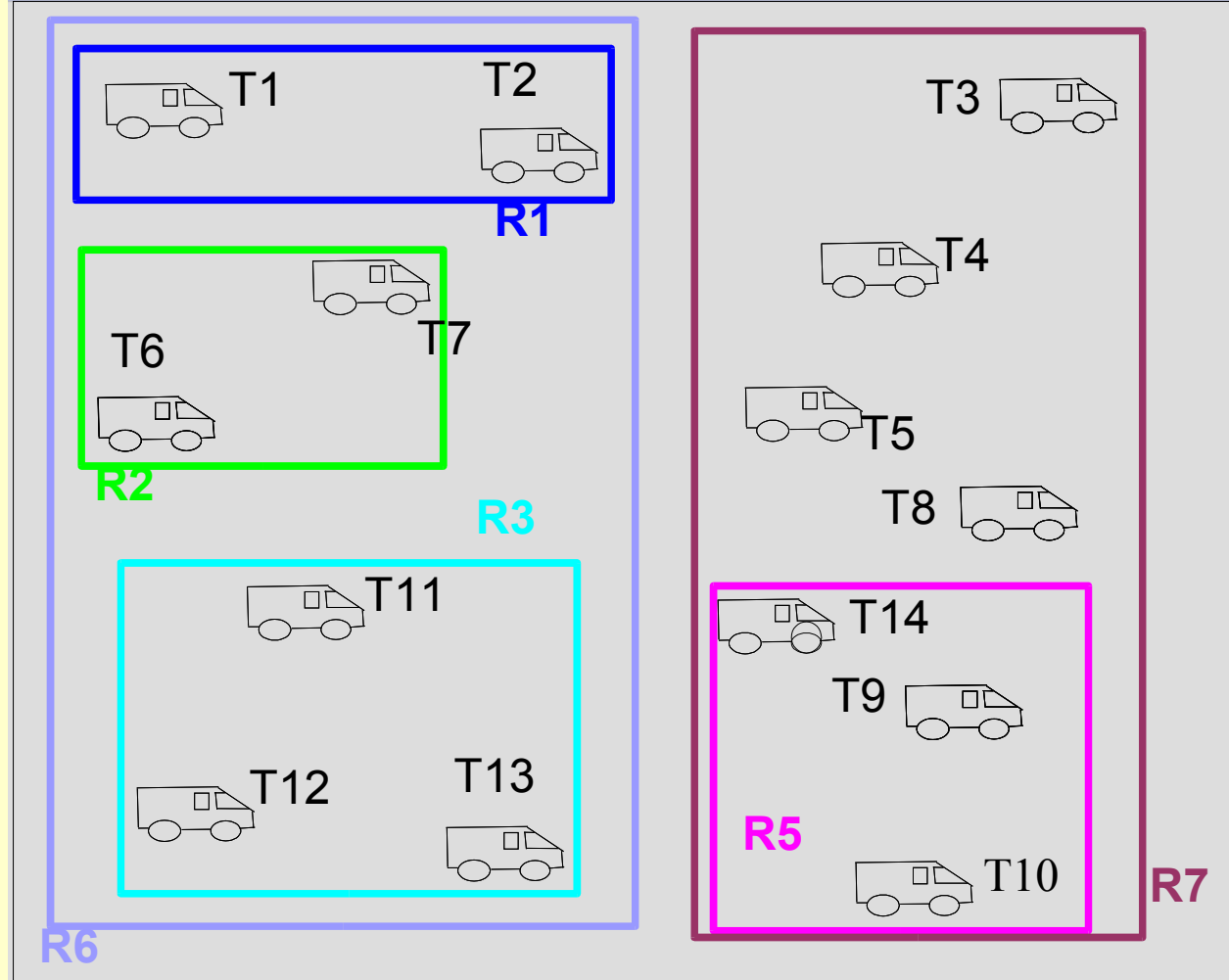
NodeSplit

NodeSplit:

ChooseAxis;

ChooseSplitIndex;

Teile auf;



x ←

Sortierung entlang der x-Achse.

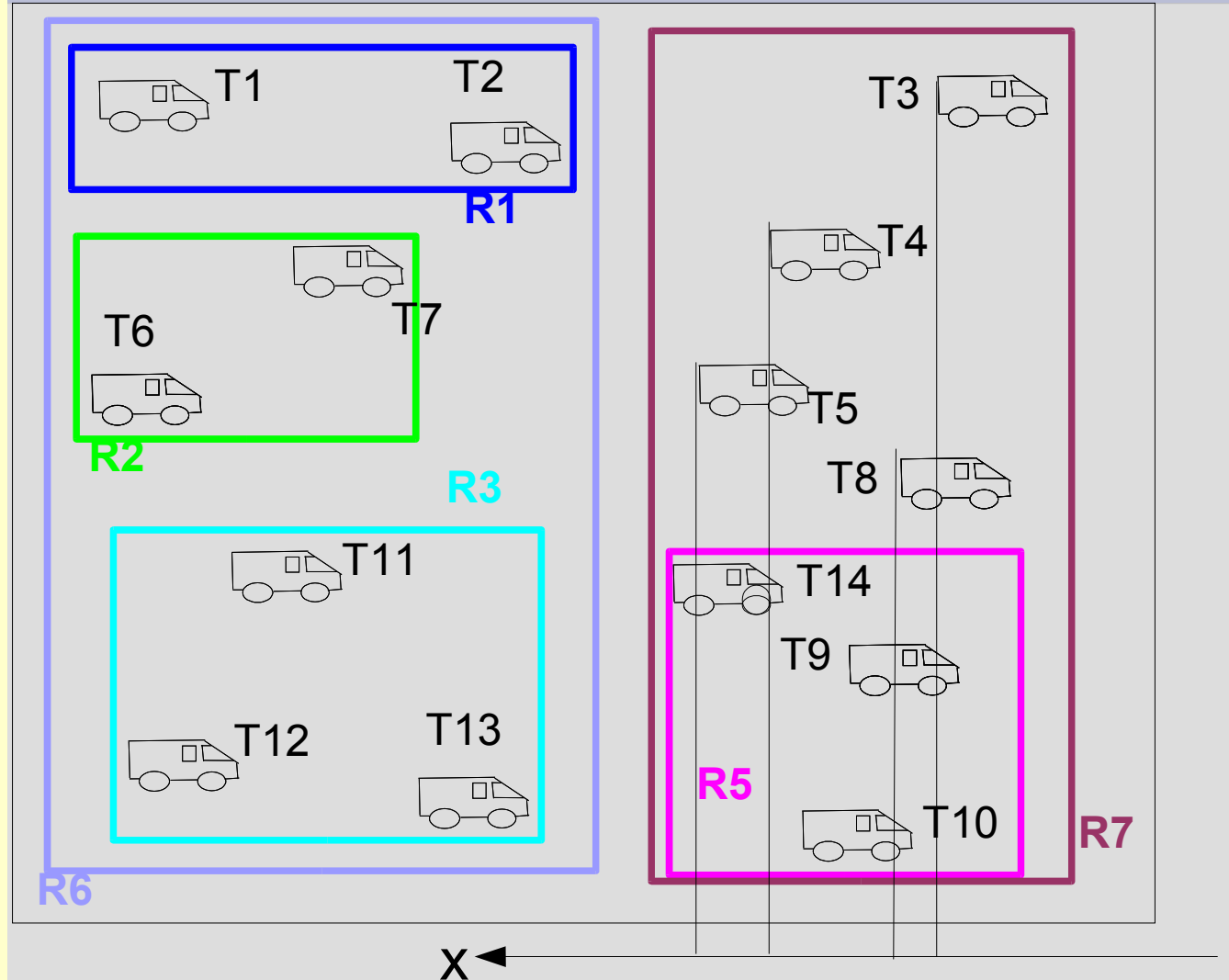
NodeSplit

NodeSplit:

ChooseAxis;

ChooseSplitIndex;

Teile auf;



Sortierung entlang der x-Achse (obere Werte).

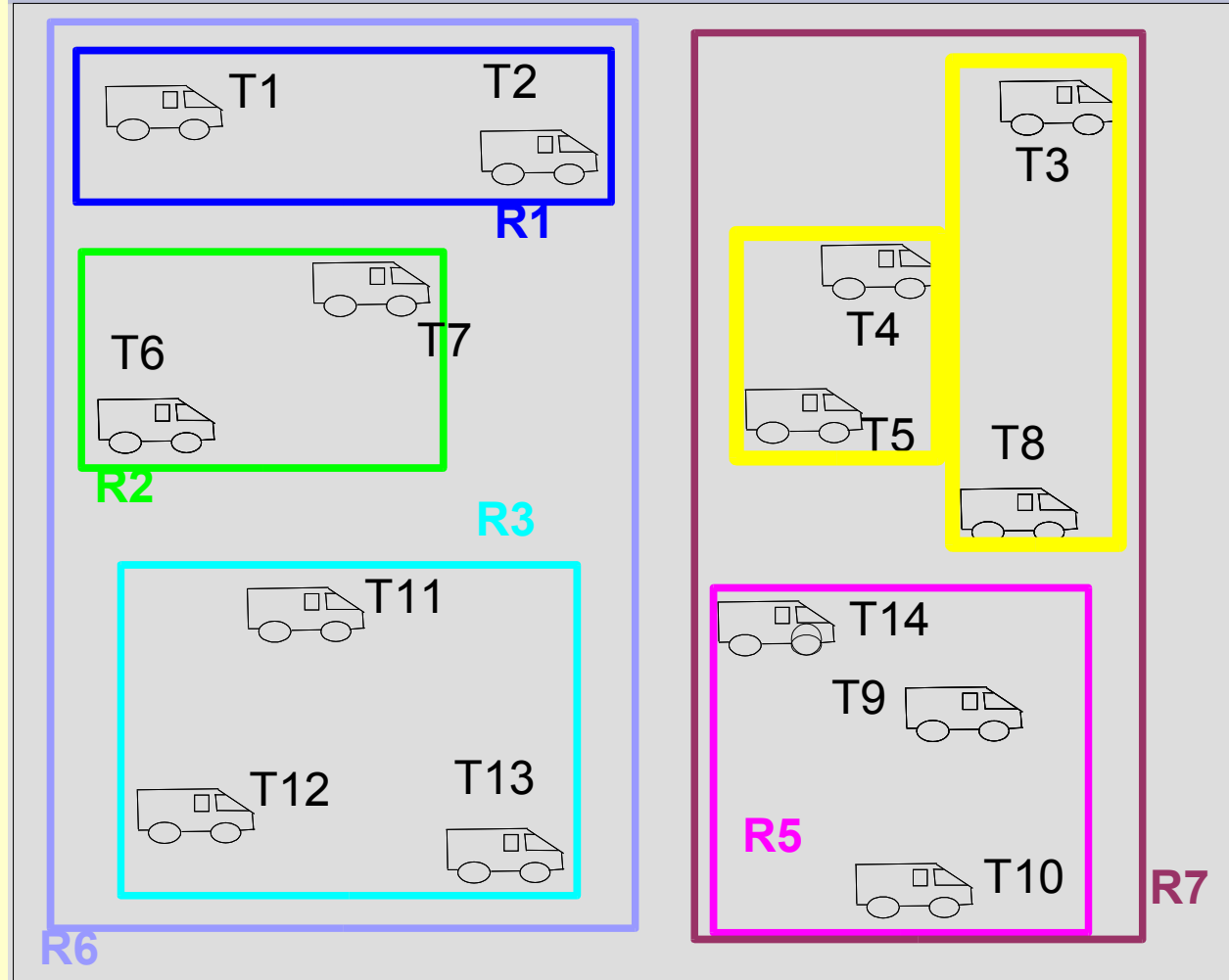
NodeSplit

NodeSplit:

ChooseAxis;

ChooseSplitIndex;

Teile auf;



x ←

Sortierung entlang der x-Achse (obere Werte).

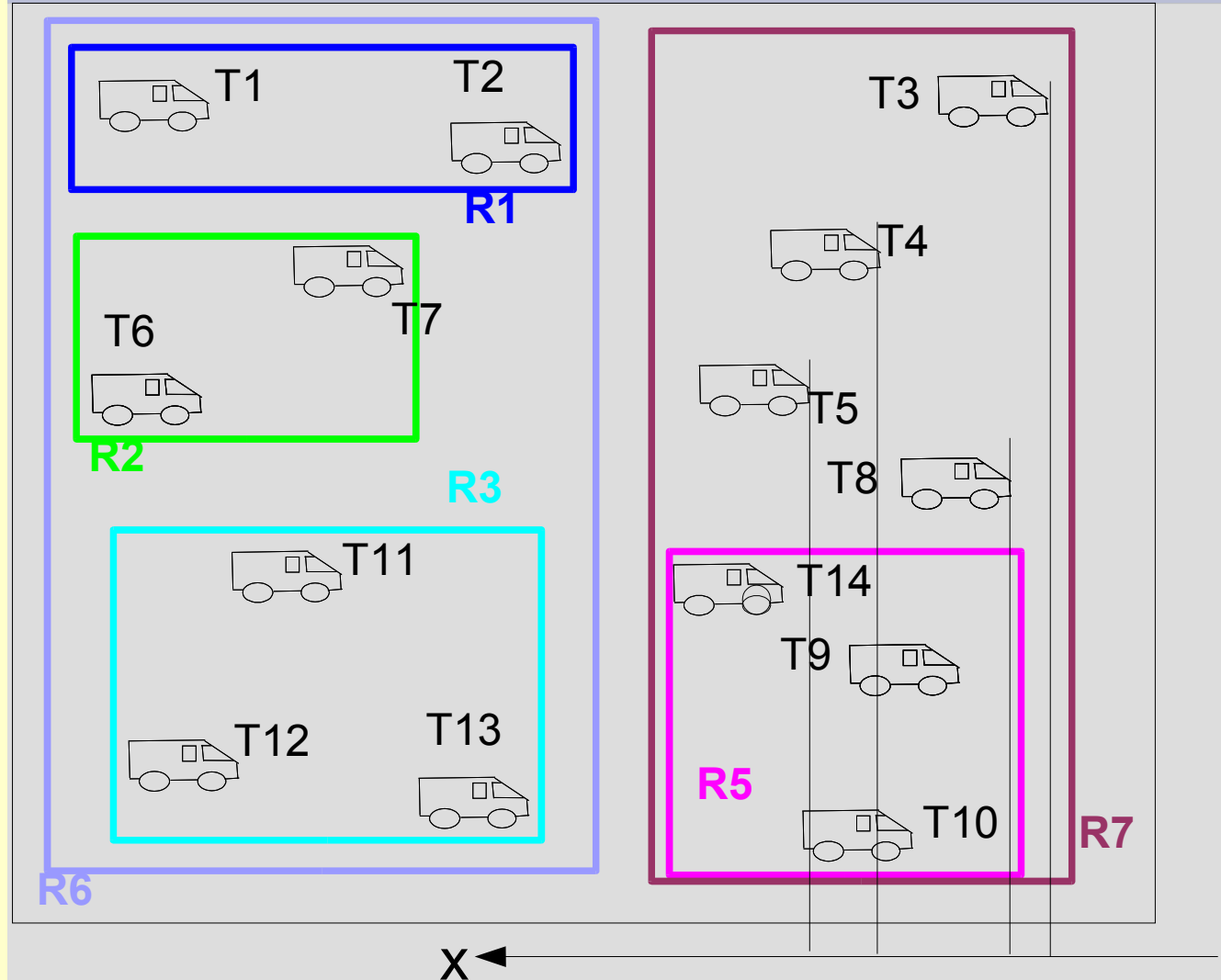
NodeSplit

NodeSplit:

ChooseAxis;

ChooseSplitIndex;

Teile auf;



Sortierung entlang der x-Achse (untere Werte)

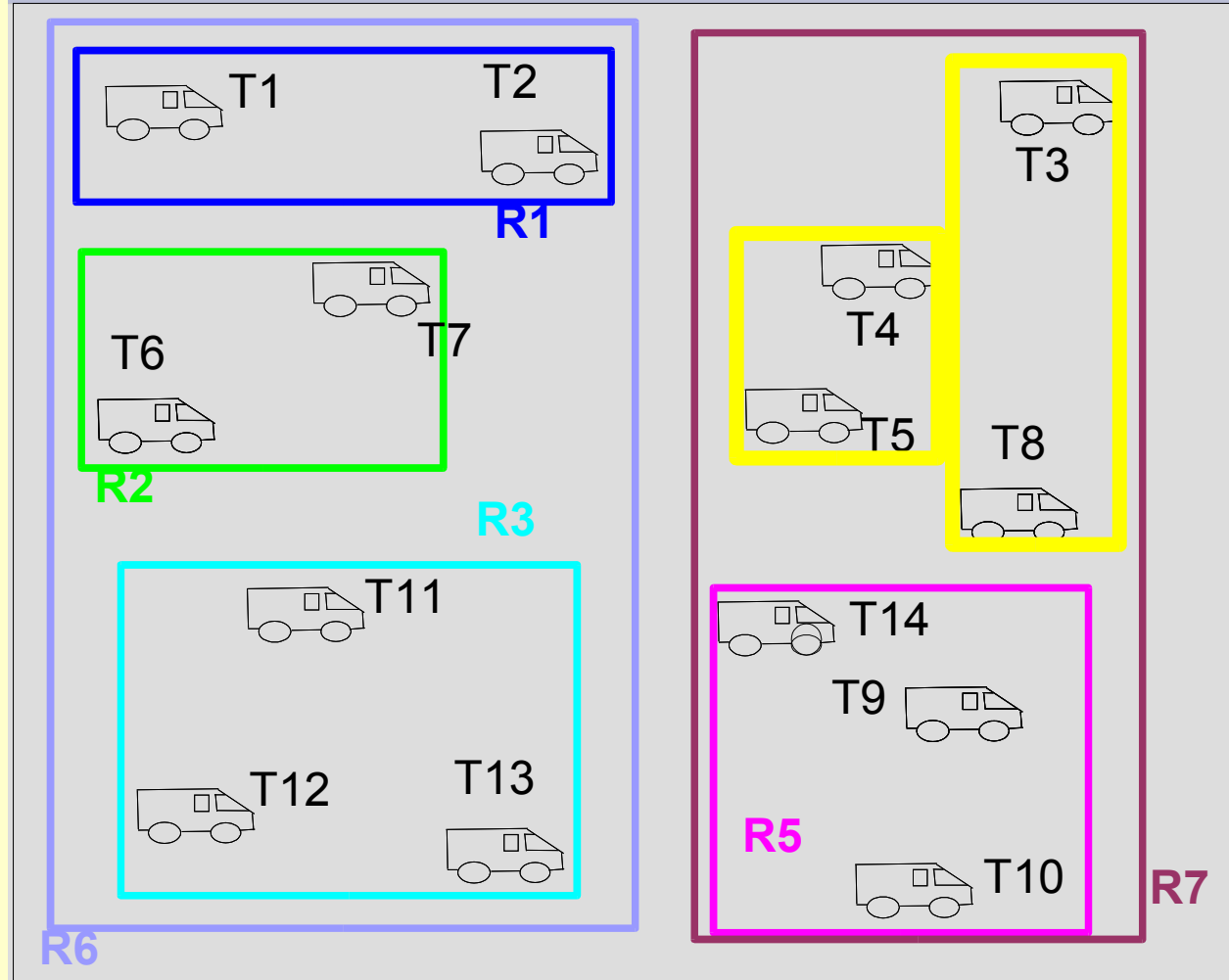
NodeSplit

NodeSplit:

ChooseAxis;

ChooseSplitIndex;

Teile auf;



x ←

Sortierung entlang der x-Achse (untere Werte).

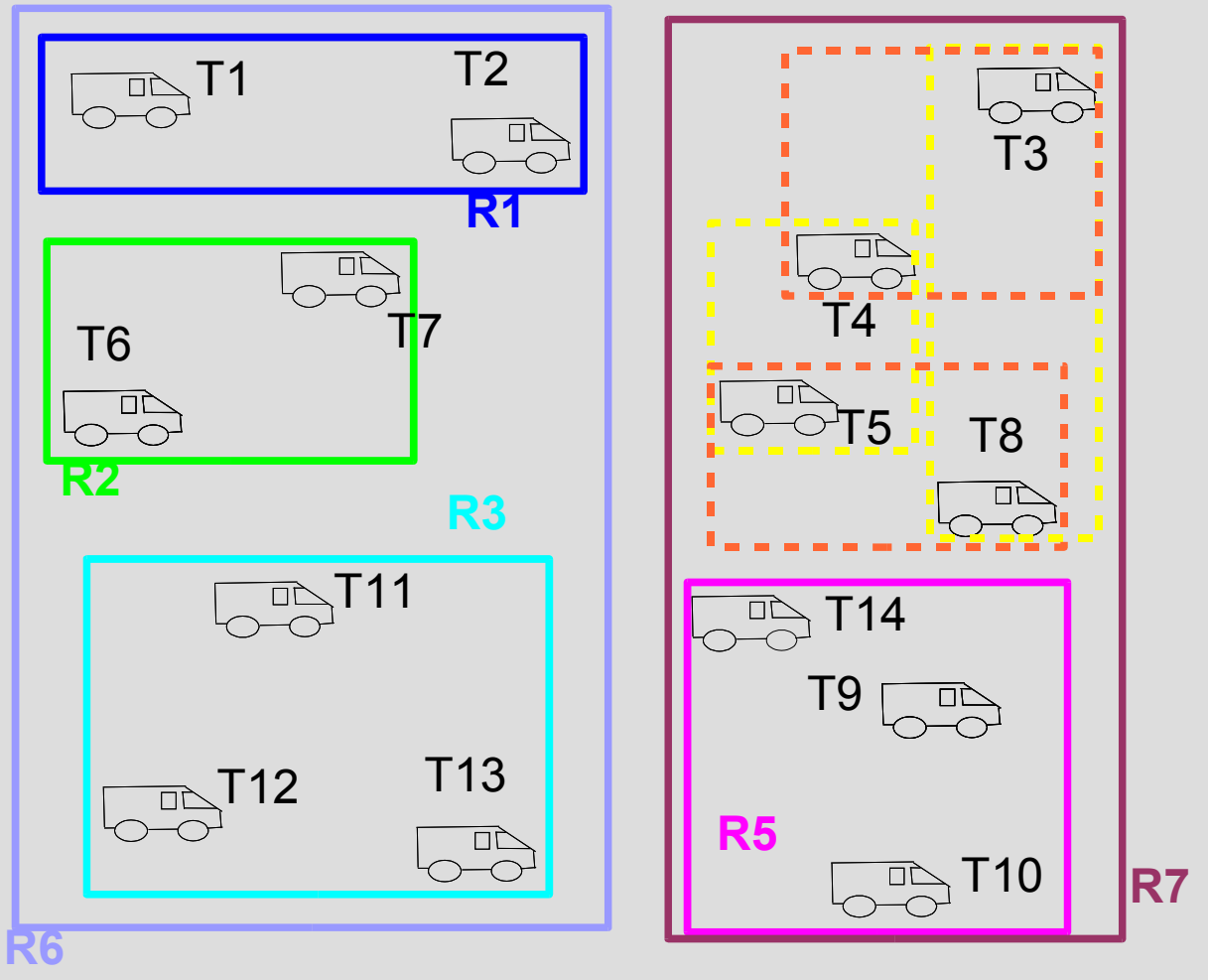
NodeSplit

NodeSplit:

ChooseAxis;

ChooseSplitIndex;

Teile auf;



Wähle Achse mit minimaler Summe der Umfangswerte.

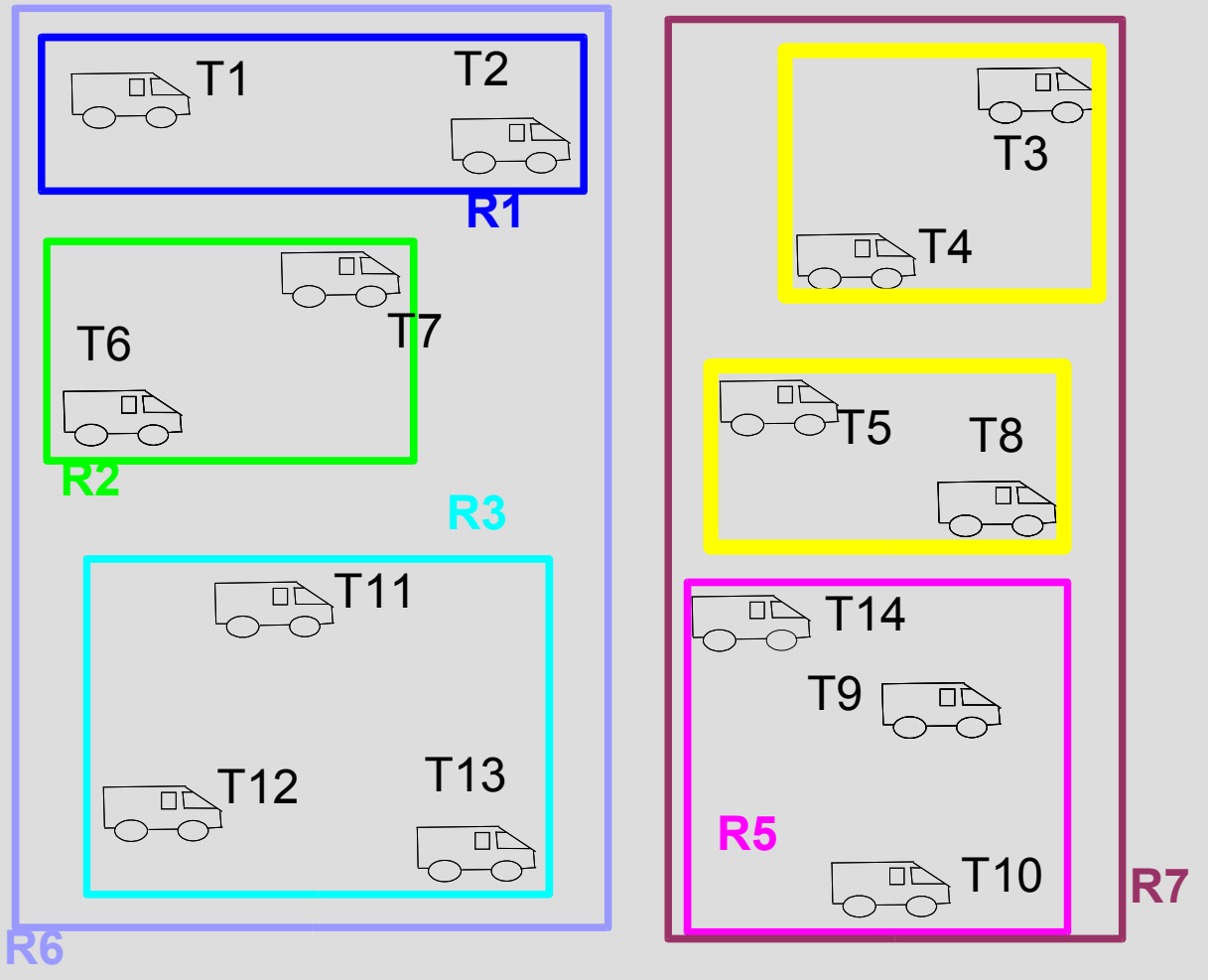
NodeSplit

NodeSplit:

ChooseAxis;

ChooseSplitIndex;

Teile auf;



Wähle Achse mit minimaler Summe der Umfangswerte: y-Achse!

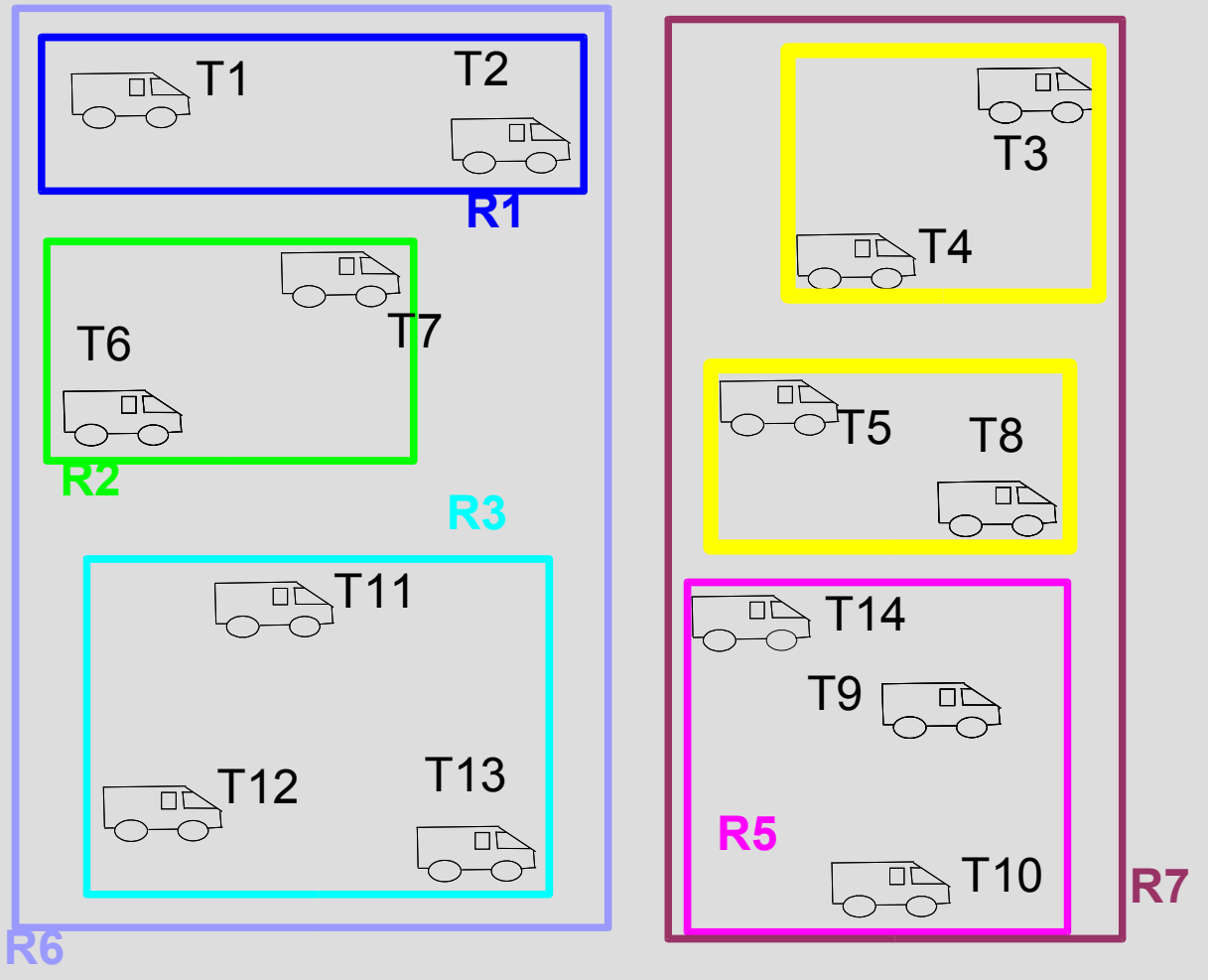
NodeSplit

NodeSplit:

ChooseAxis;

ChooseSplitIndex;

Teile auf;

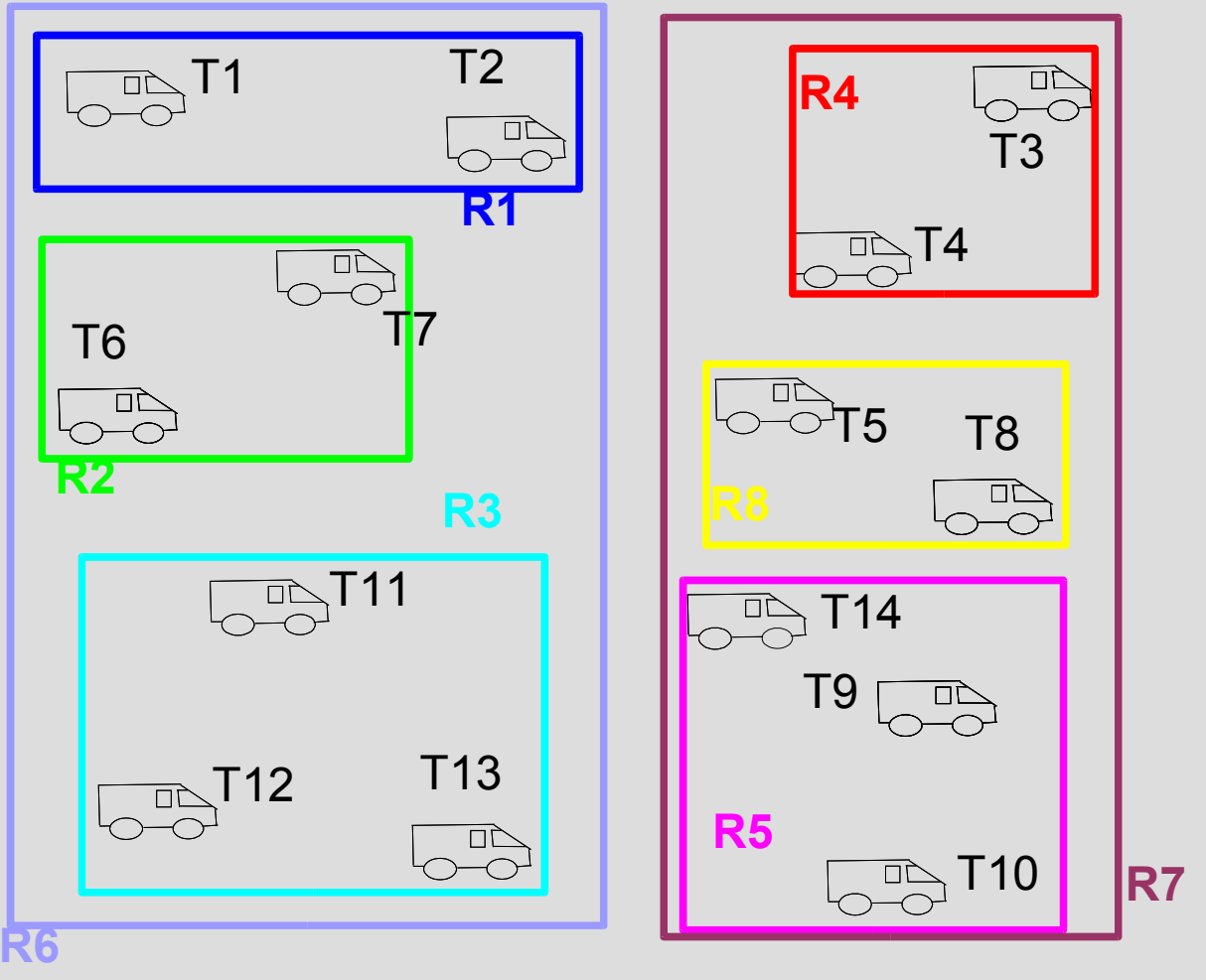


Wähle Verteilung mit **minimalem Überlappungswert.**

NodeSplit

NodeSplit:

ChooseAxis;
ChooseSplitIndex;
Teile auf;



Fertig.

Indexstrukturen

- B-Baum
 - keine effizienten räumlichen Suchanfragen möglich
- R-Baum
 - Erweiterung des B-Baum (mehrdimensionale Indexierung)
- R*-Baum
 - „Performance-optimierter“ R-Baum
- TPR-Baum
 - Anpassung des R*-Baum für „sich stetig fortbewegende Objekte“

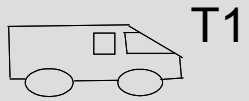
Mobile Objekte?

- Bisherige Annahme
 - ▶ Daten bleiben unverändert, bis sie explizit geändert werden
- Problem: sich gleichmäßig fortbewegende Objekte
 - ▶ sehr viele Änderungen
 - ▶ veraltete, falsche Daten

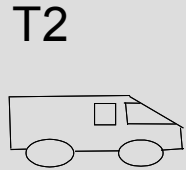
Mobile Objekte?

- Welcher Lieferwagen befindet sich **innerhalb der nächsten 5 min** in der Nähe (Umkreis von 2 km) vom Informatikgebäude am Fasanengarten?
 - Welchen anderen Lieferwagenfahrer werde ich **in den nächsten 30 min** treffen (um gemeinsam Pause zu machen)?
- **Anfragen, die sich auf die Zukunft beziehen, können mit bisherigen Strukturen nicht beantwortet werden!**

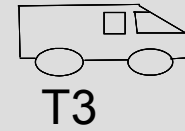
Stehende Transporter



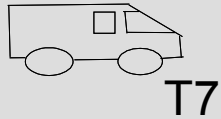
T1



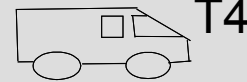
T2



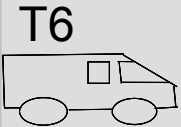
T3



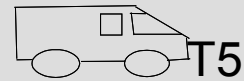
T7



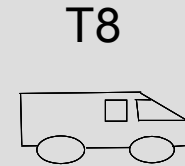
T4



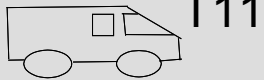
T6



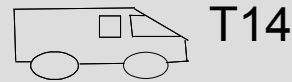
T5



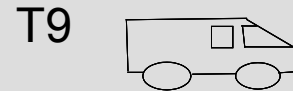
T8



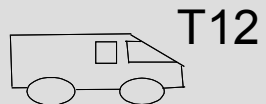
T11



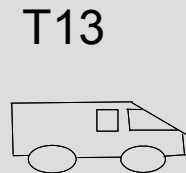
T14



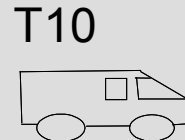
T9



T12

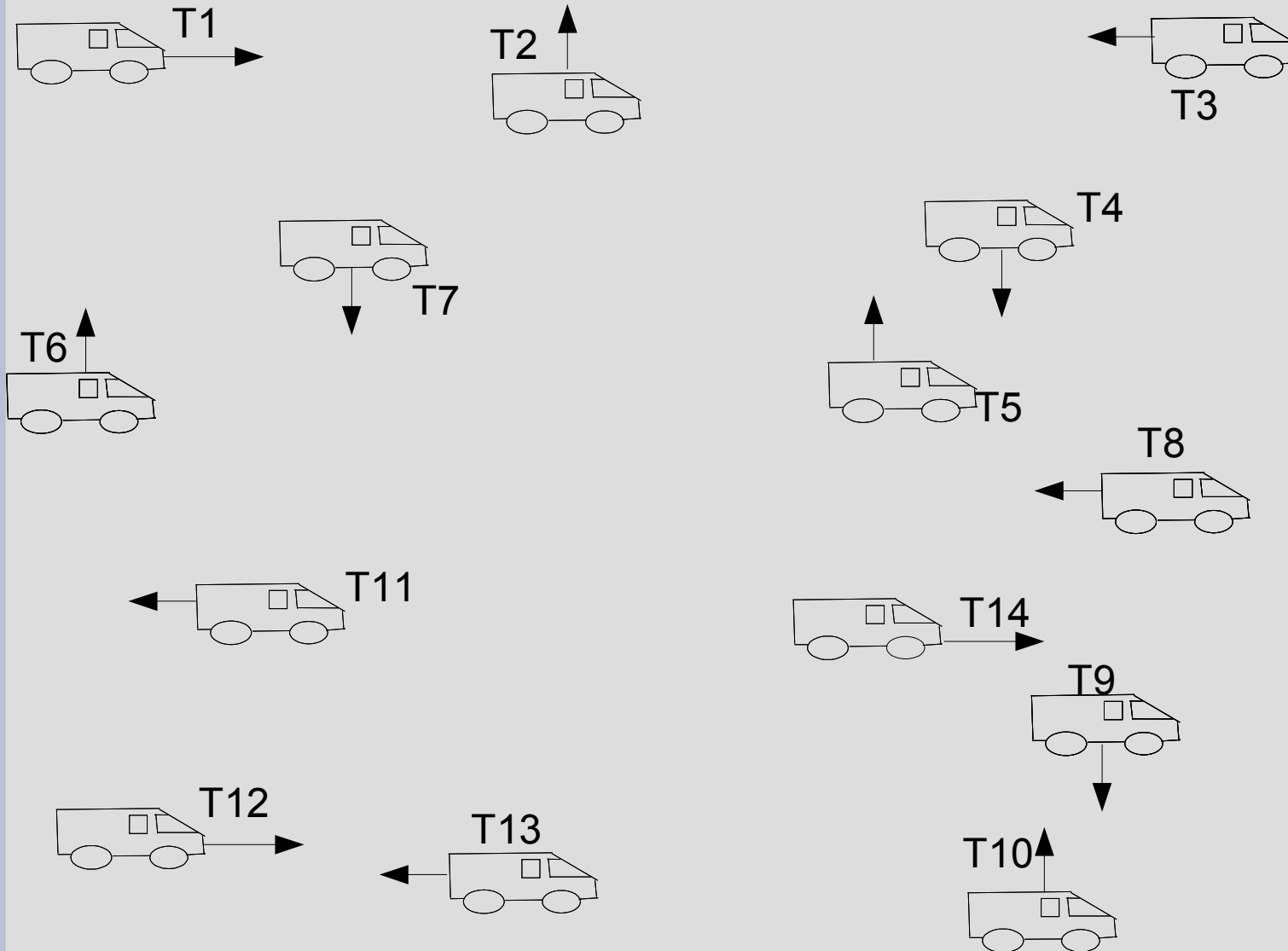


T13

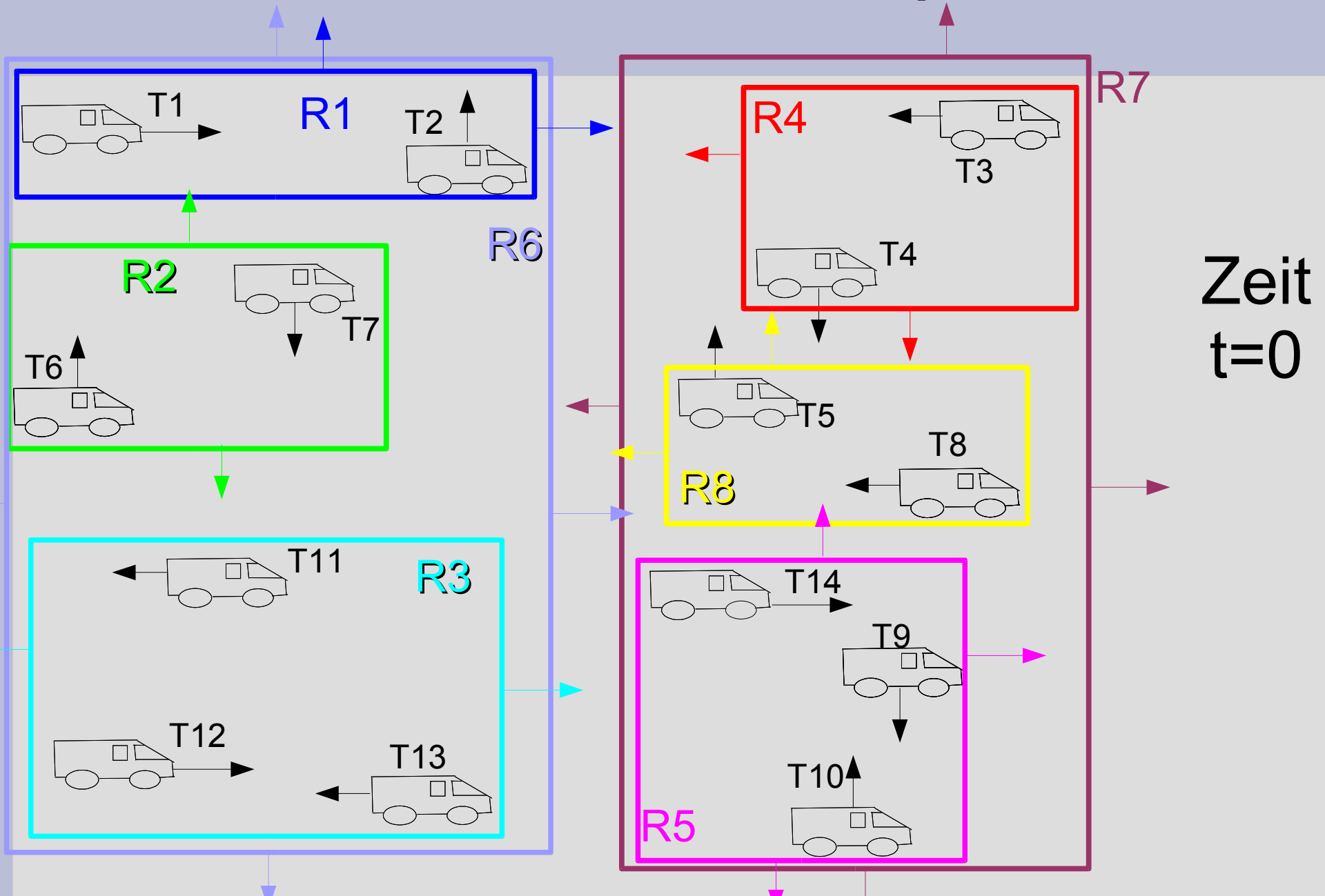


T10

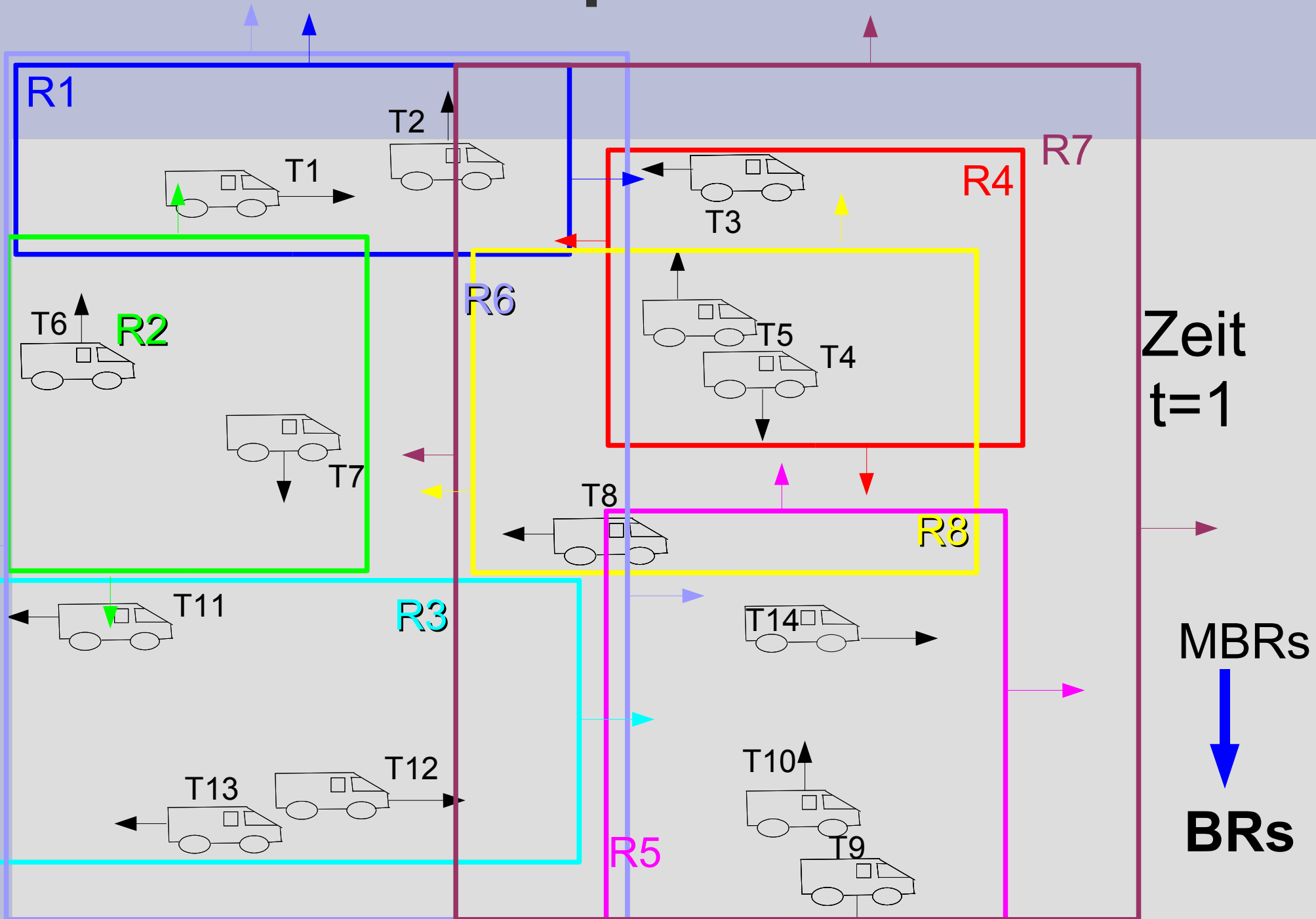
Fahrende Transporter



Fahrende Transporter im Time Parameterized R-Baum (TPR-Baum)



Fahrende Transporter im TPR-Baum

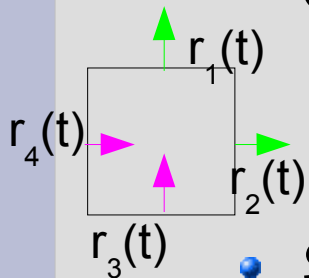


TPR-Baum

Objekte werden als Funktionen gespeichert:
 $x(t) = x(t_0) + v(t - t_0)$ mit $t \geq \text{jetzt}$

→ BRs werden als Funktionen gespeichert:

- für jede Seite eine Funktion $r_i(t)$:
 - Seite lässt BR "wachsen":
 - Position = äußerste Position eines Elements im BR
 - Geschwindigkeit = **größte** Geschw. eines Element im BR
 - Seite lässt BR "schrumpfen":
 - Position = äußerste Position eines Element im BR
 - Geschwindigkeit = **kleinste** Geschw. eines Element im BR



TPR-Baum

- Erweiterung des R^* -Baums
- Algorithmen vom R^* -Baum werden übernommen, wobei die Funktionen über die Eigenschaften von BRs durch ihre Integrale über die Zeit ersetzt werden

NodeSplit

zusätzlich:

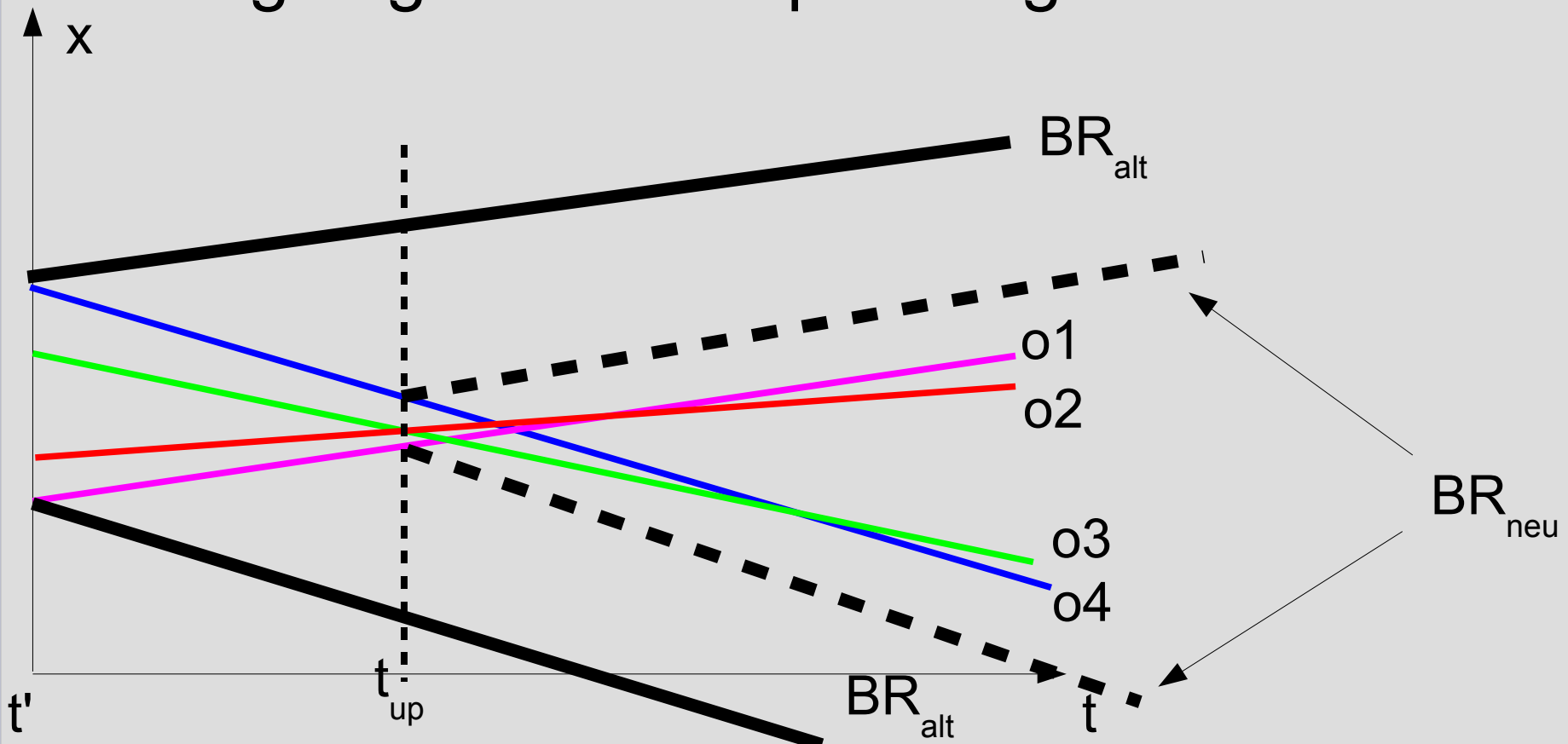
Sortierung der Geschwindigkeitsvektoren

- geringere Geschwindigkeitsunterschiede
- geringeres Wachstum des BR

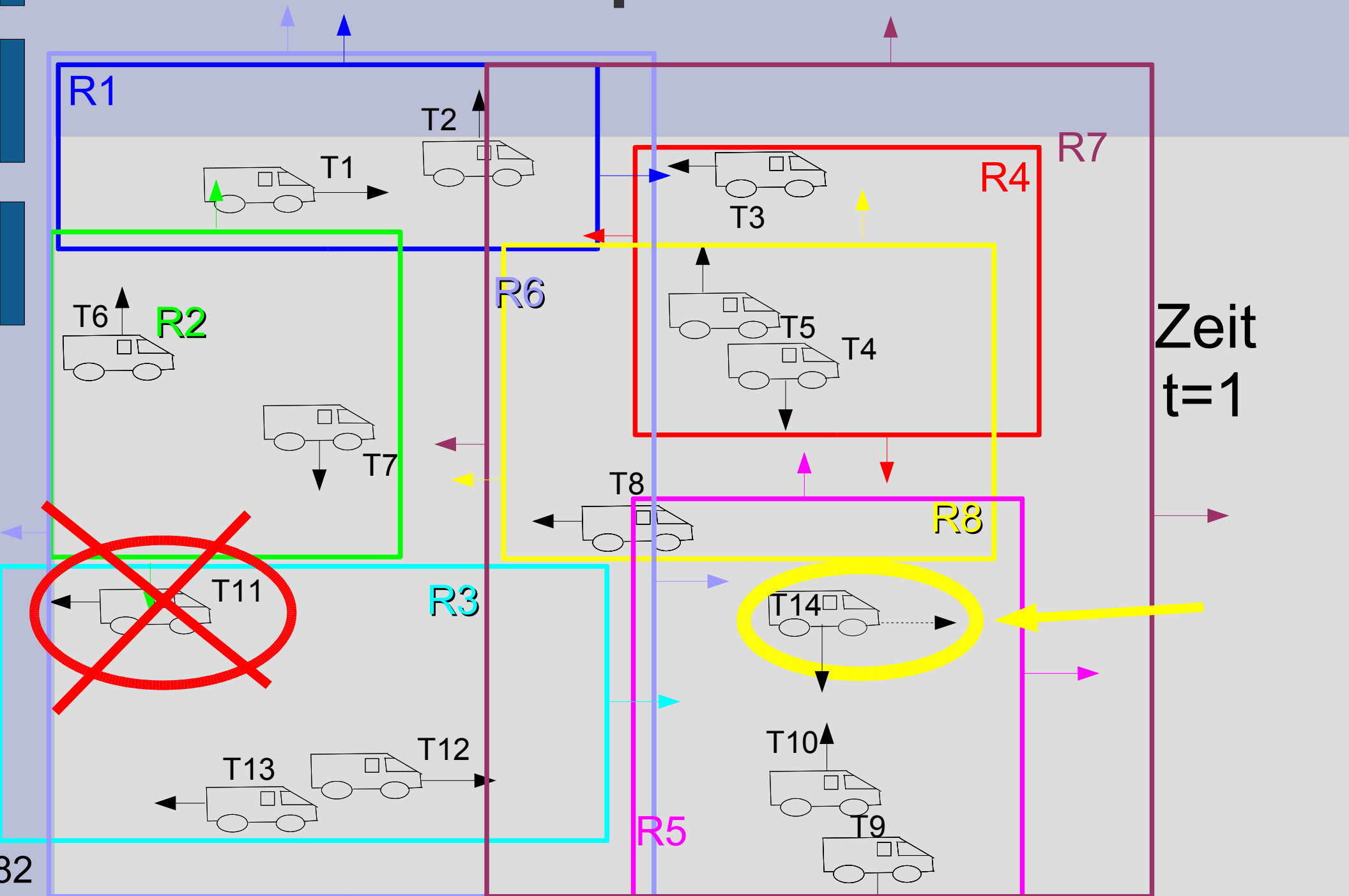
TPR-Baum

- BRs schrumpfen nie

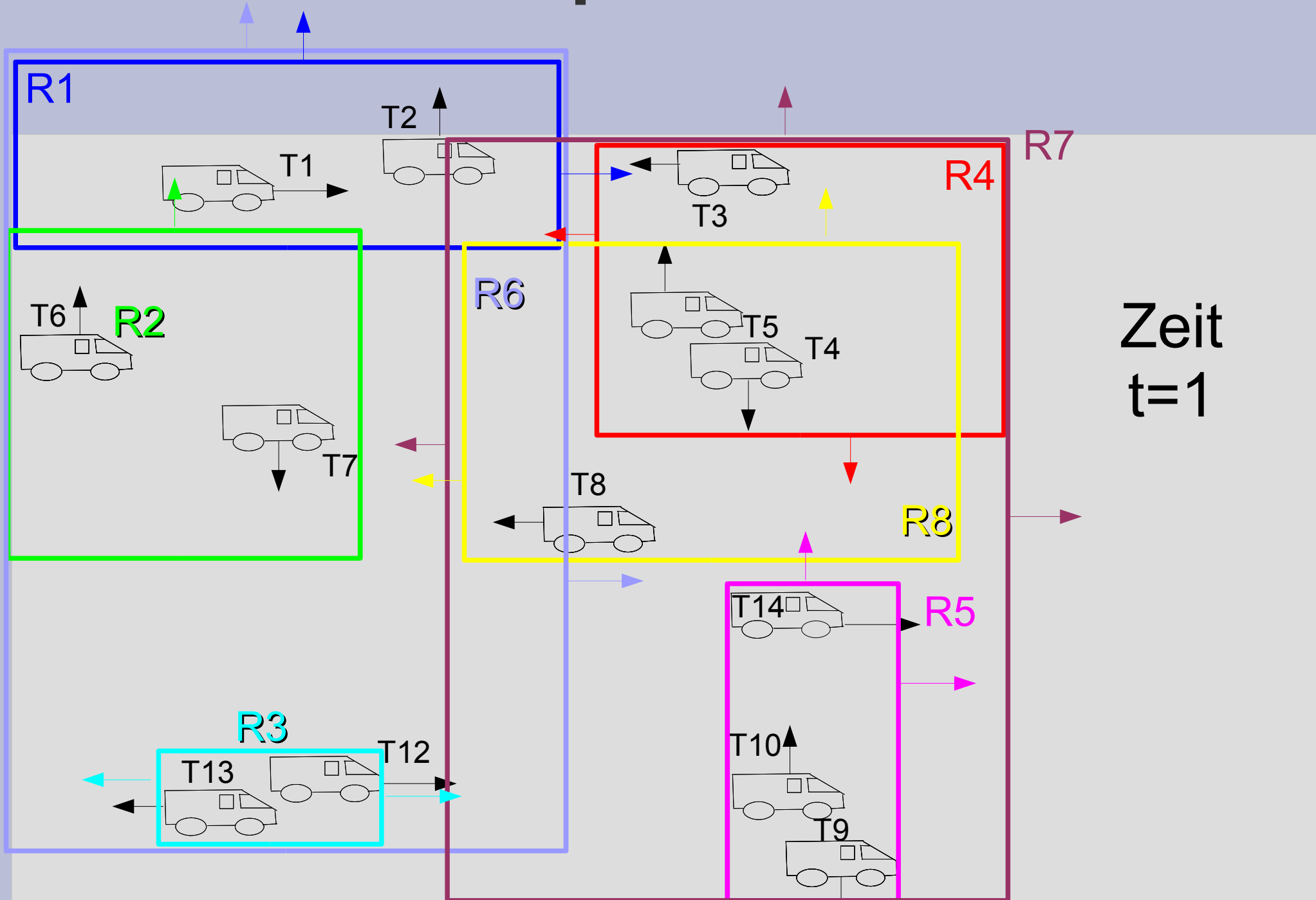
 gelegentliche Anpassungen



Fahrende Transporter im TPR-Baum



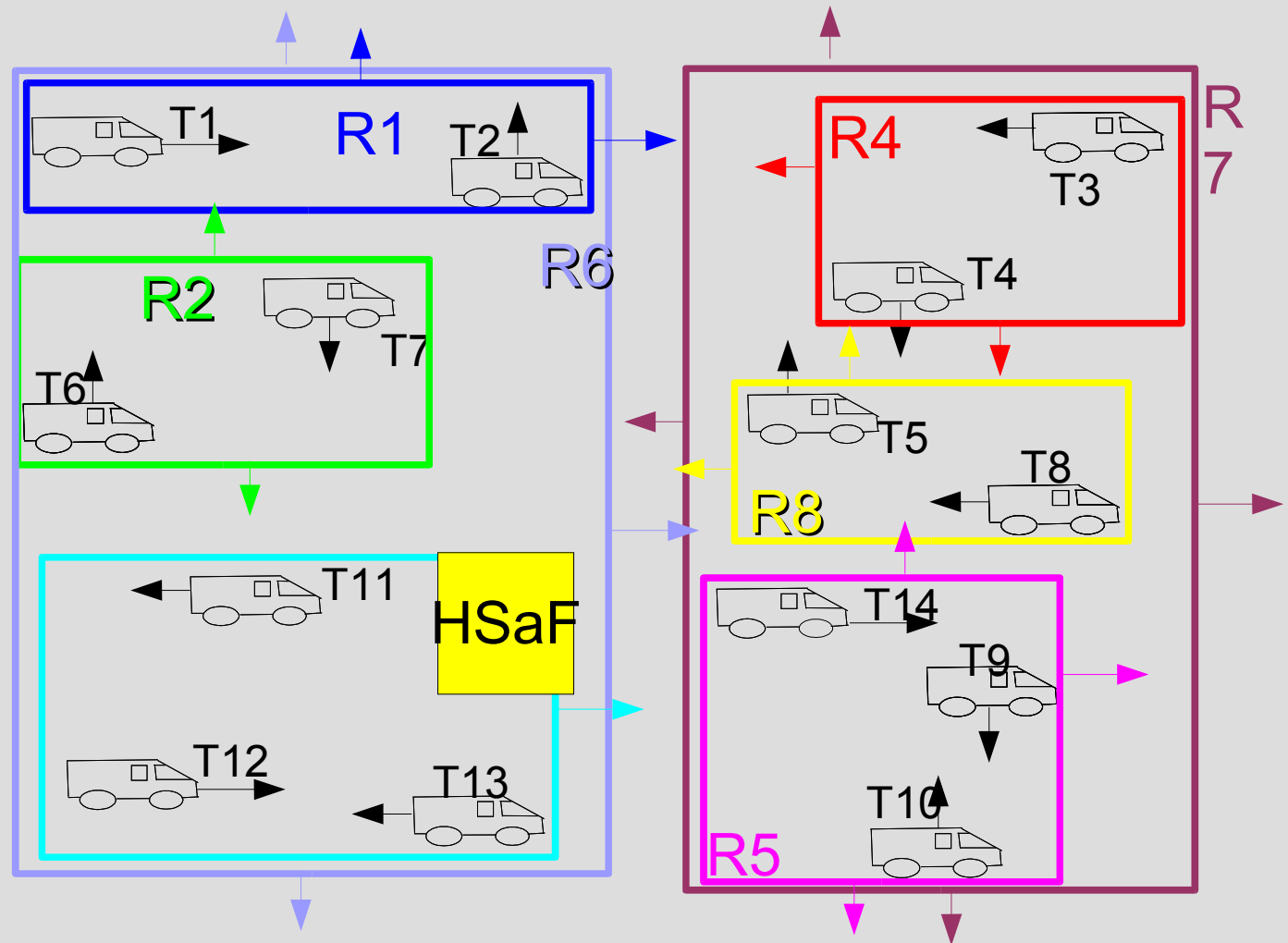
Fahrende Transporter im TPR-Baum



Anfragen

Welcher
Transporter
befindet sich in
der Nähe des
HSaF?

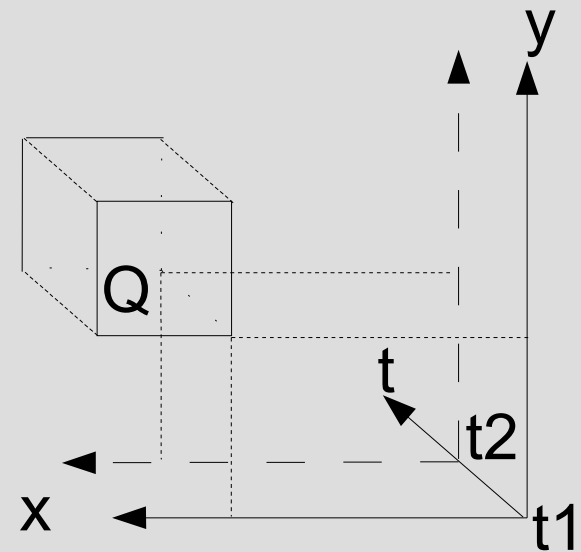
Wird **wie im R-
Baum** bearbeitet.
Unterschied:
Die **Rechtecke**
müssen für den
Anfragezeitpunkt
berechnet
werden.



Anfragen

1. Welcher Lieferwagen befindet sich innerhalb der nächsten 5 min in der Nähe (Umkreis von 2 km) vom Informatikgebäude am Fasanengarten?

Anfragefenster Q wird zu einem sich **bewegenden** Fenster. Es bewegt sich **entlang der Zeitachse**.

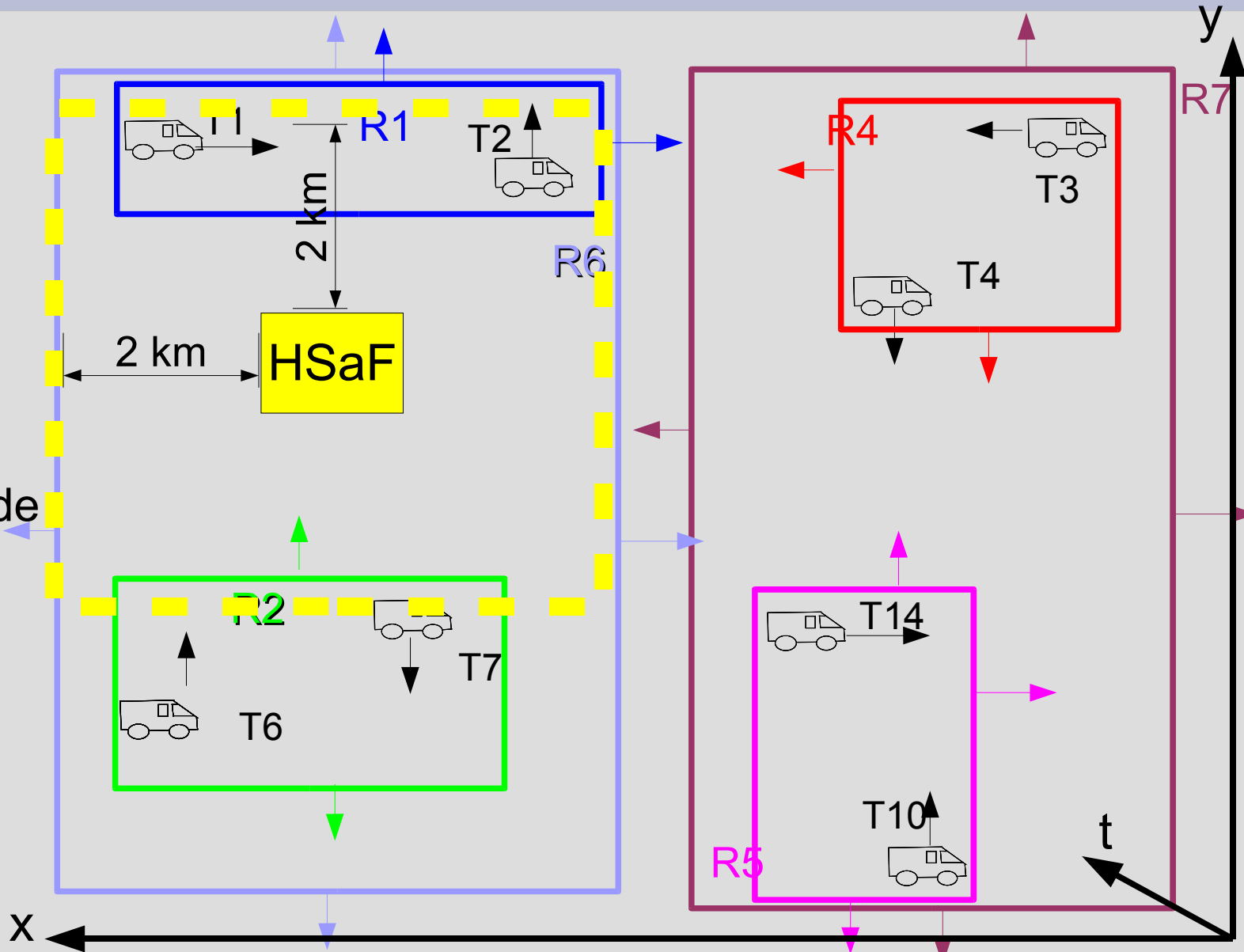


Anfragen

Anfrage 1

Welcher Lieferwagen befindet sich innerhalb der nächsten 5 min in der Nähe (Umkreis von 2 km) vom Informatikgebäude am Fasanengarten?

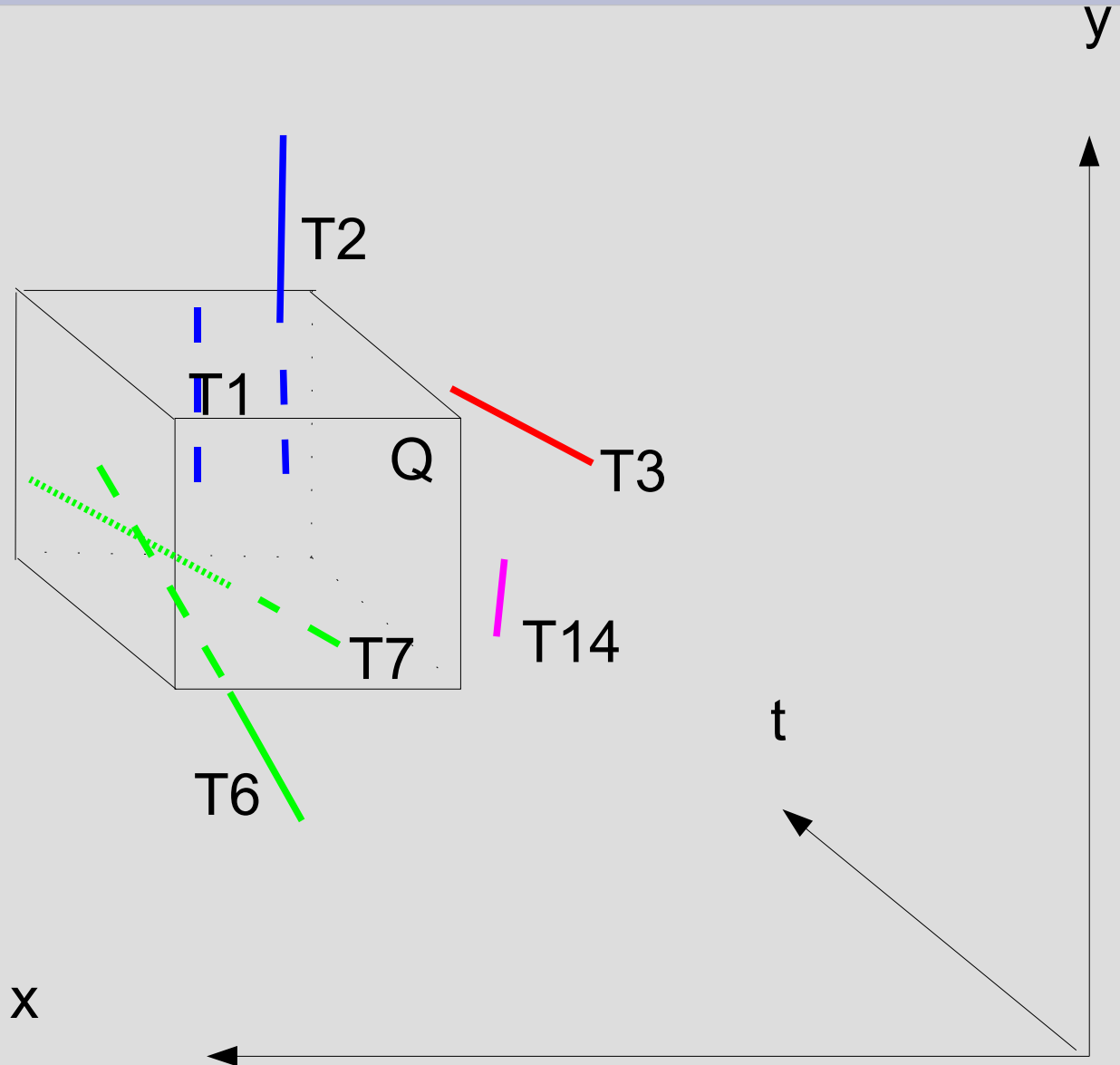
Zeit:
 t
(in Minuten)



Anfragen

Anfrage 1

Welcher Lieferwagen befindet sich innerhalb der nächsten 5 min in der Nähe (Umkreis von 2 km) vom Informatikgebäude am Fasanengarten?

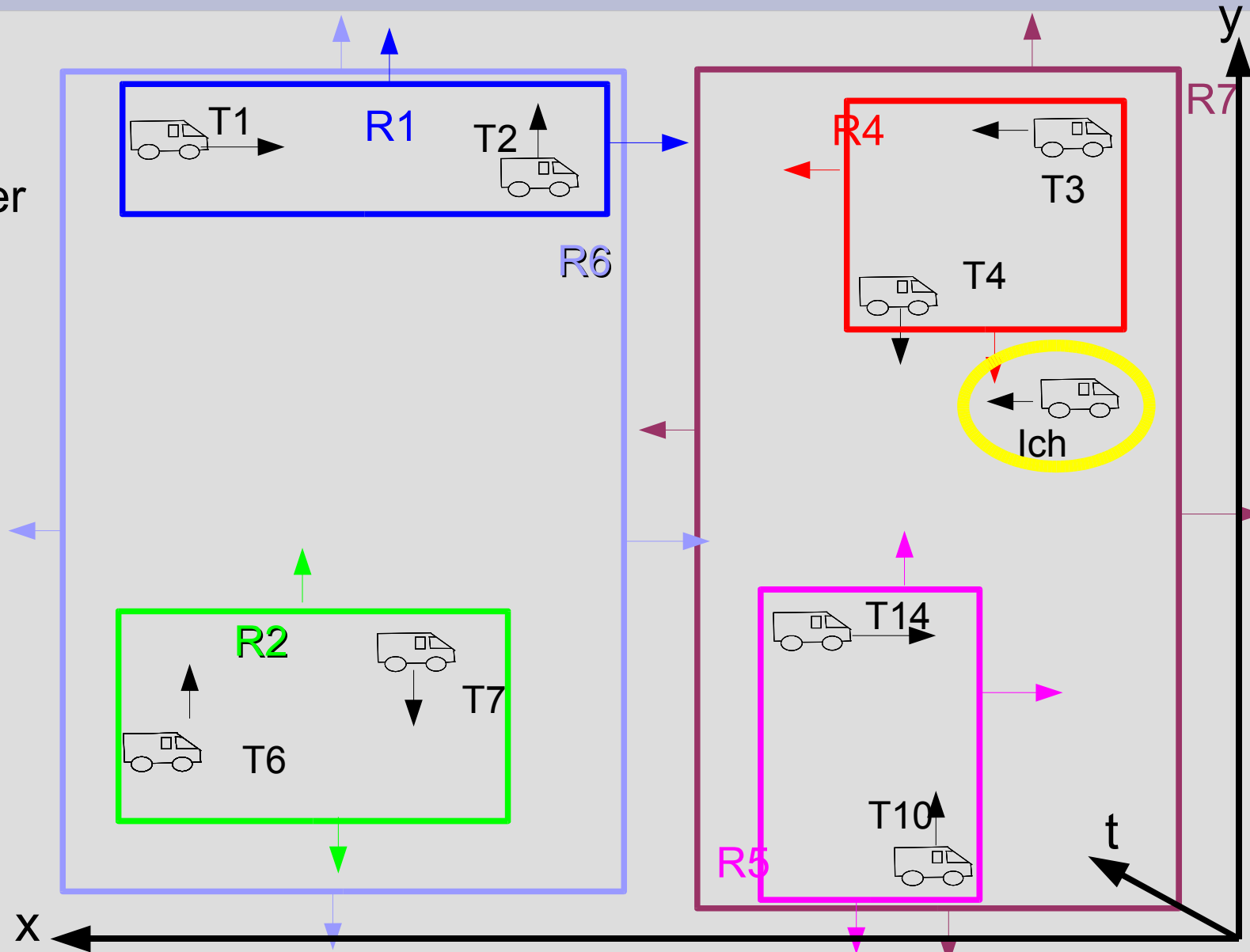


Anfragen

2. Welchen anderen Lieferwagenfahrer werde ich in den nächsten 30 Minuten treffen (um gemeinsam Pause zu machen)?

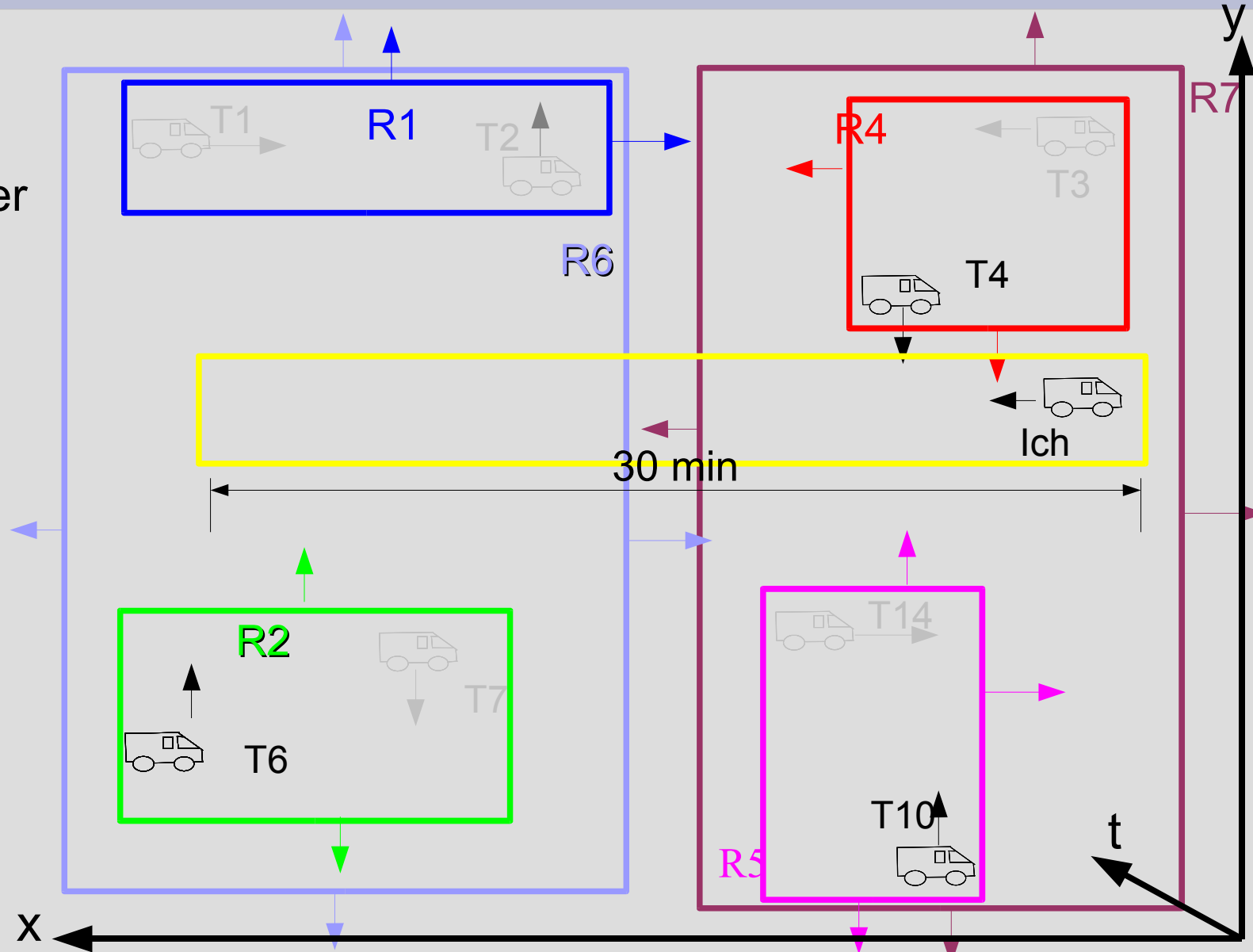
Anfragen

Anfrage 2
Welchen
anderen
Lieferwagenfahrer
werde ich in den
nächsten 30 min
treffen (um
gemeinsam
Pause
zu machen)?



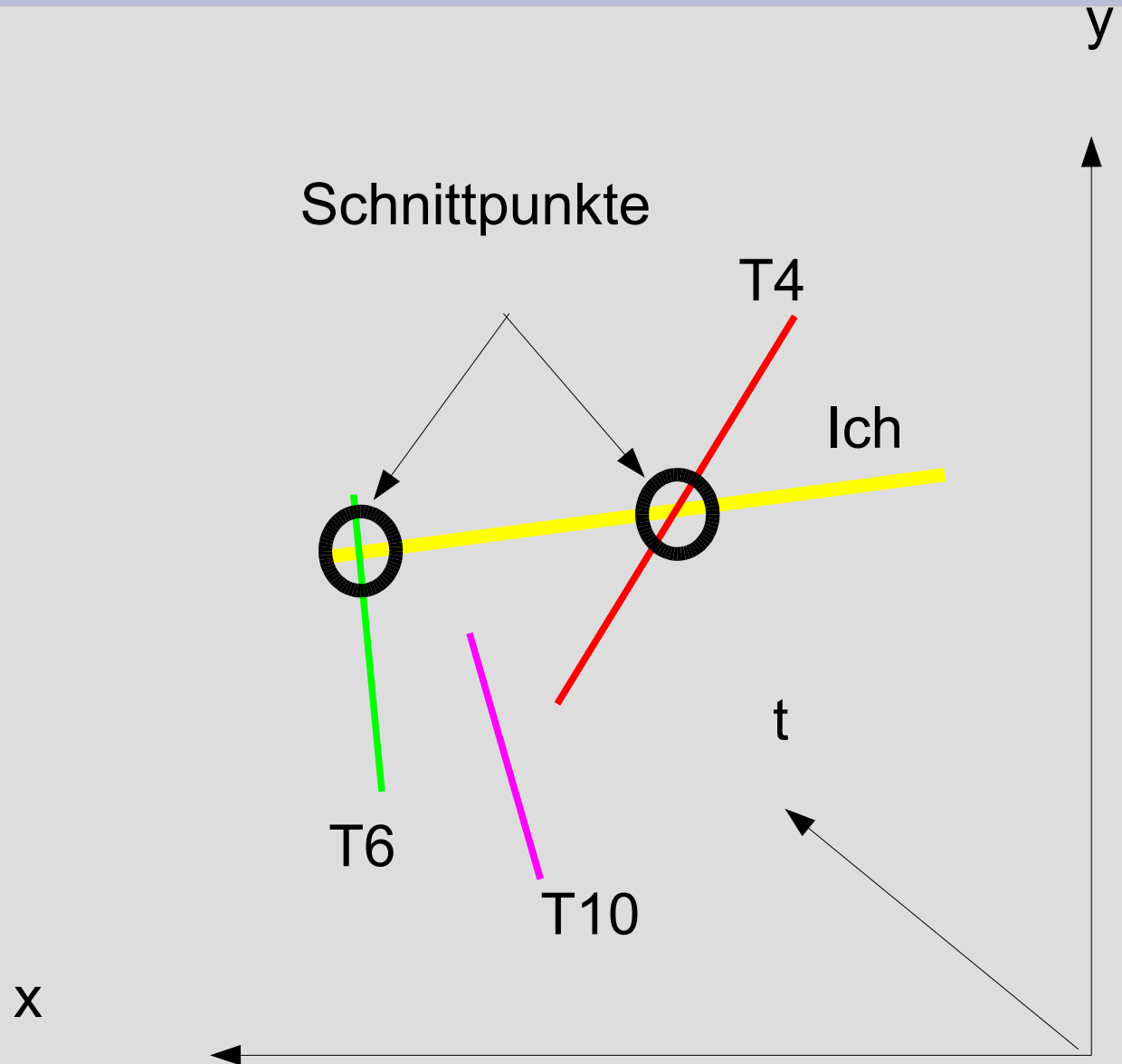
Anfragen

Anfrage 2
Welchen
anderen
Lieferwagenfahrer
werde ich in den
nächsten 30 min
treffen (um
gemeinsam
Pause
zu machen)?



Anfragen

Anfrage 2
Welchen
anderen
Lieferwagenfahrer
werde ich in den
nächsten 30 min
treffen (um
gemeinsam
Pause
zu machen)?



Anfragen

Anfrage 3

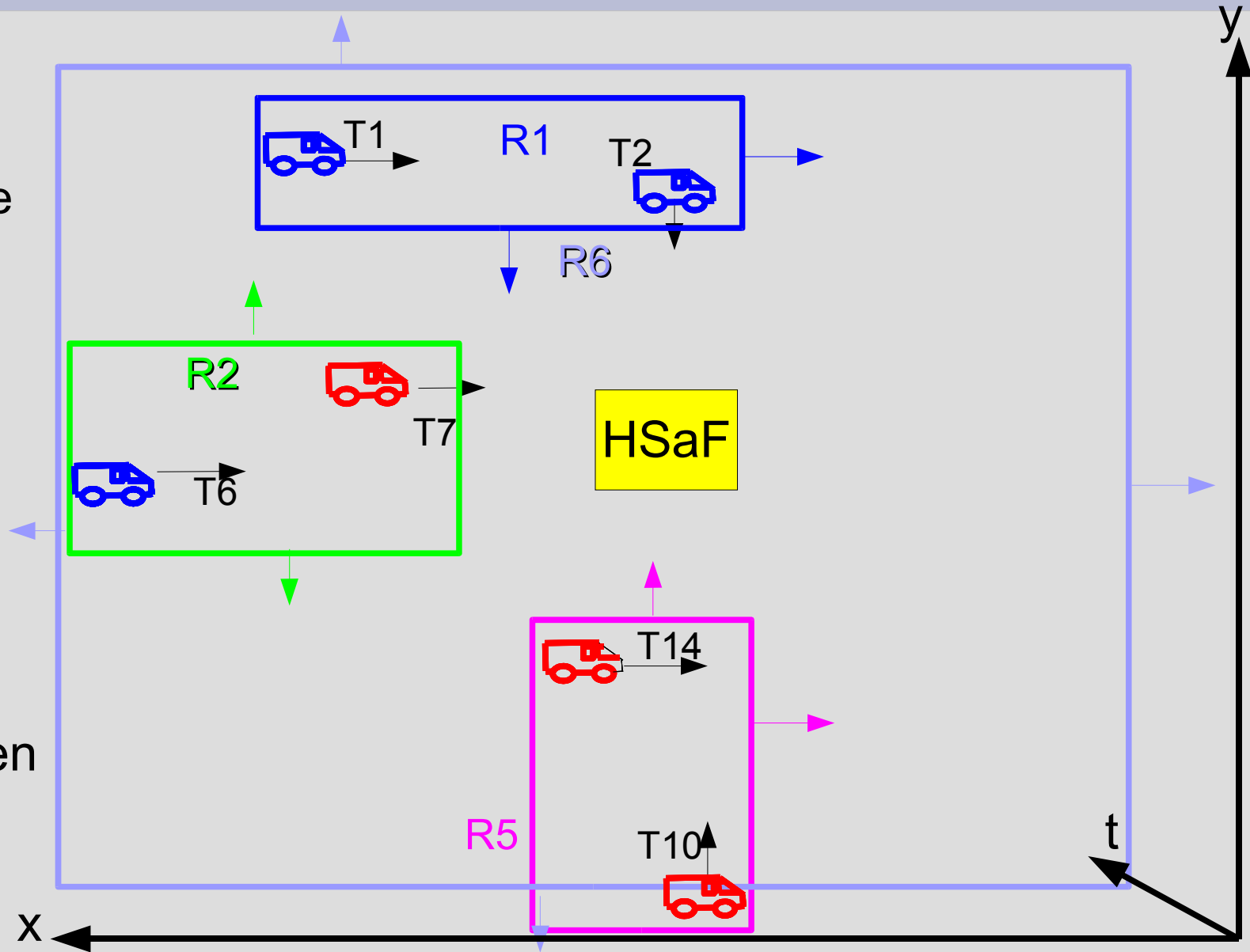
Welche Lieferwagen sind unbeladen bis sie den Fasanengarten erreichen?



beladen



unbeladen



Anfragen

Anfrage 3

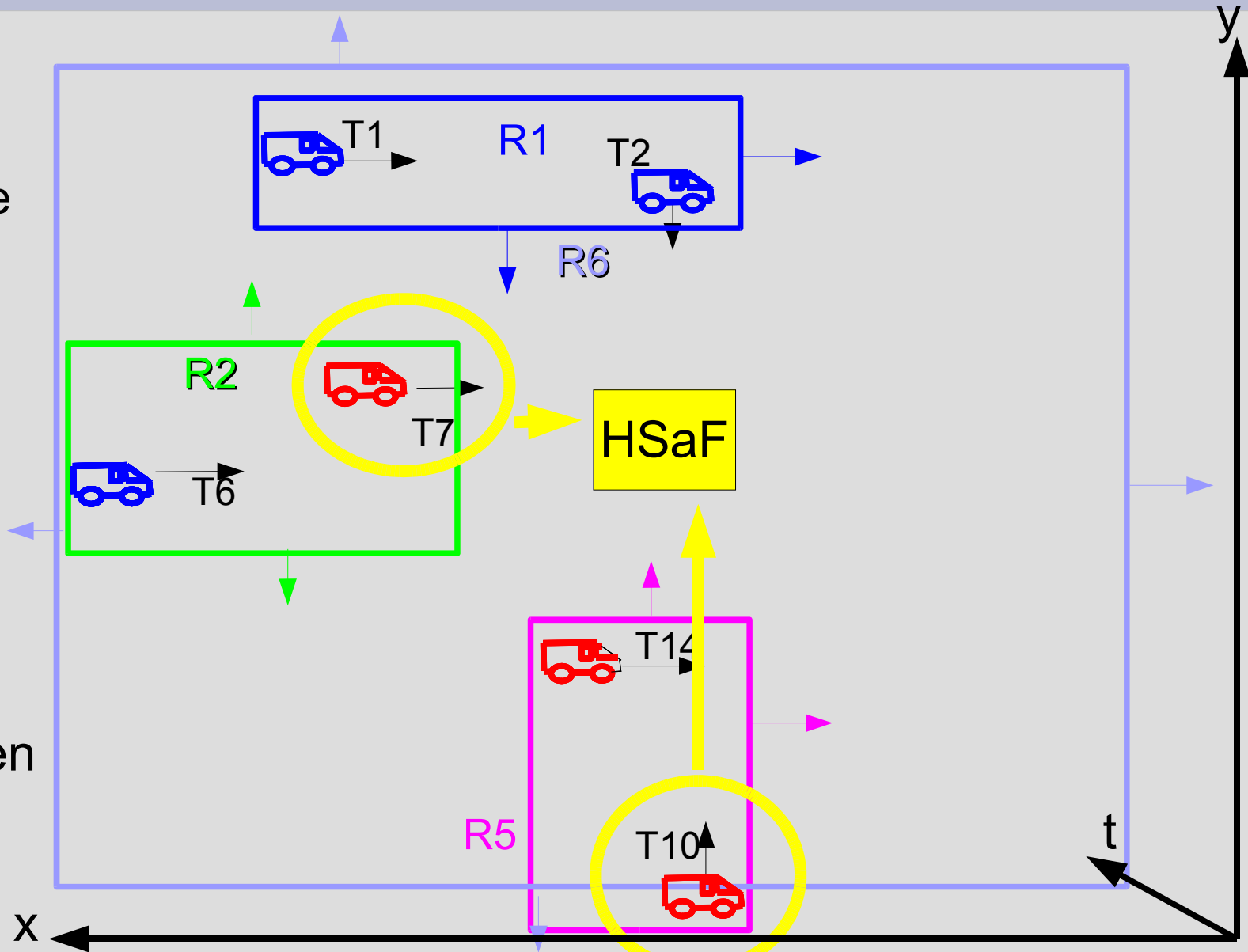
Welche Lieferwagen sind unbeladen bis sie den Fasanengarten erreichen?



beladen



unbeladen



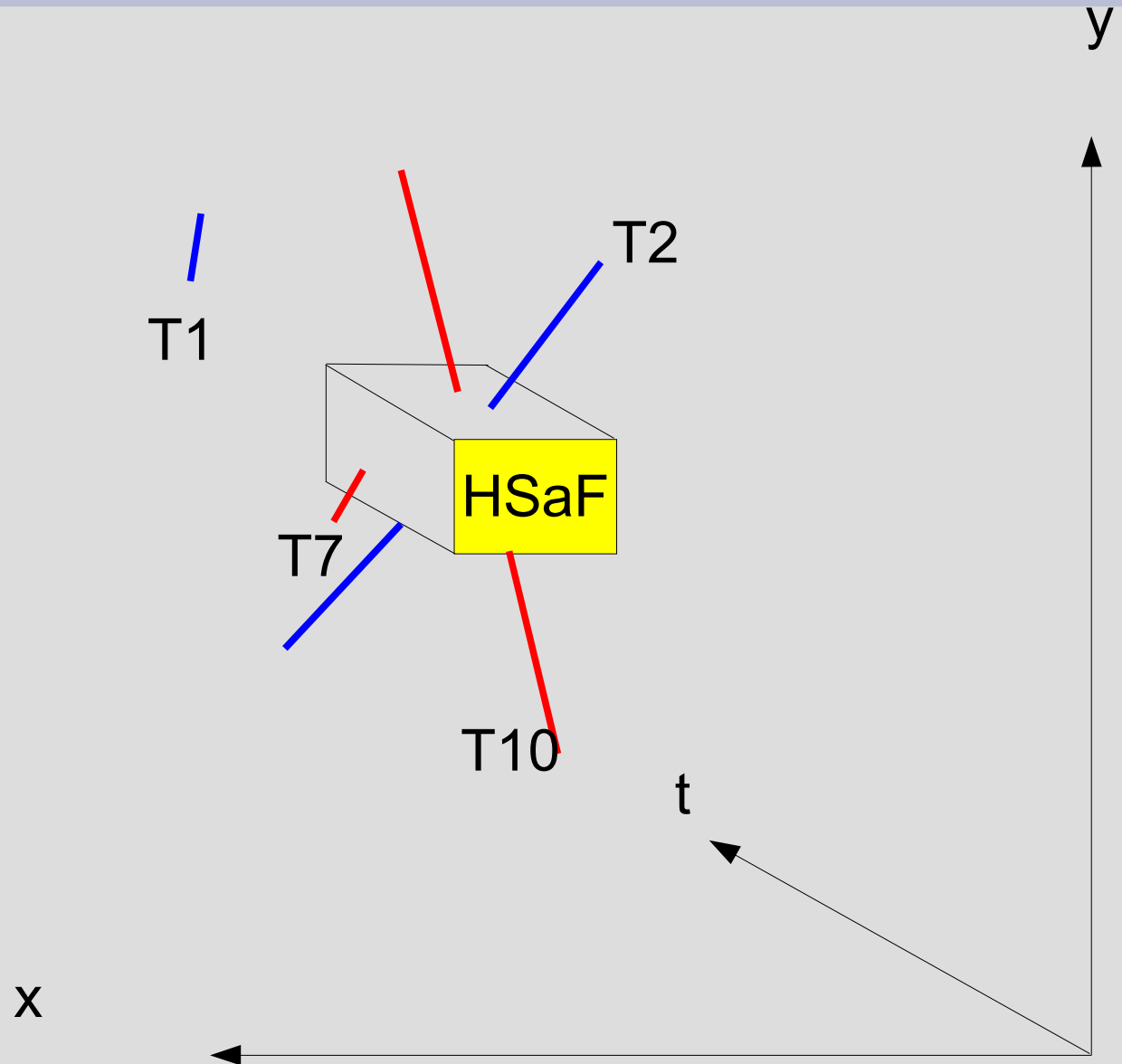
Anfragen

Anfrage 3

Welche Lieferwagen sind unbeladen bis sie den Fasanengarten erreichen?

— beladen

— unbeladen



Anfragen

- Je weiter die Anfrage in die Zukunft reicht, desto schlechter wird die Qualität der Antwort

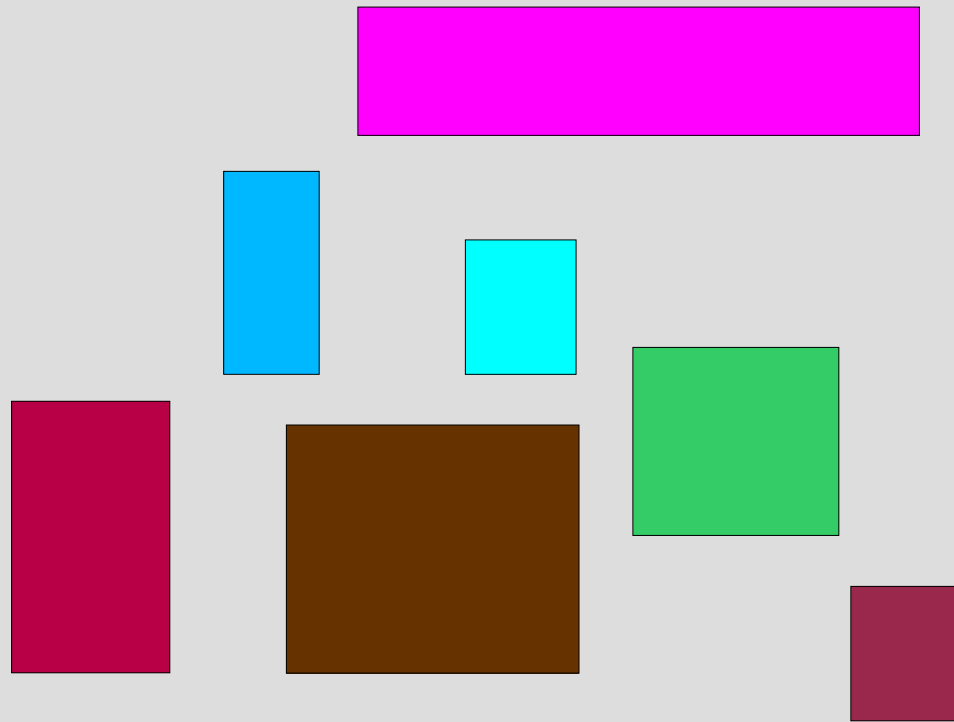
→ Unsicherheit

Indexstrukturen

- B-Baum
 - keine effizienten räumlichen Suchanfragen möglich
- R-Baum
 - Erweiterung des B-Baum (mehrdimensionale Indexierung)
- R*-Baum
 - „Performance-optimierter“ R-Baum
- TPR-Baum
 - Anpassung des R*-Baum für „sich stetig fortbewegende Objekte“

Ausschlußregionen

Idee:



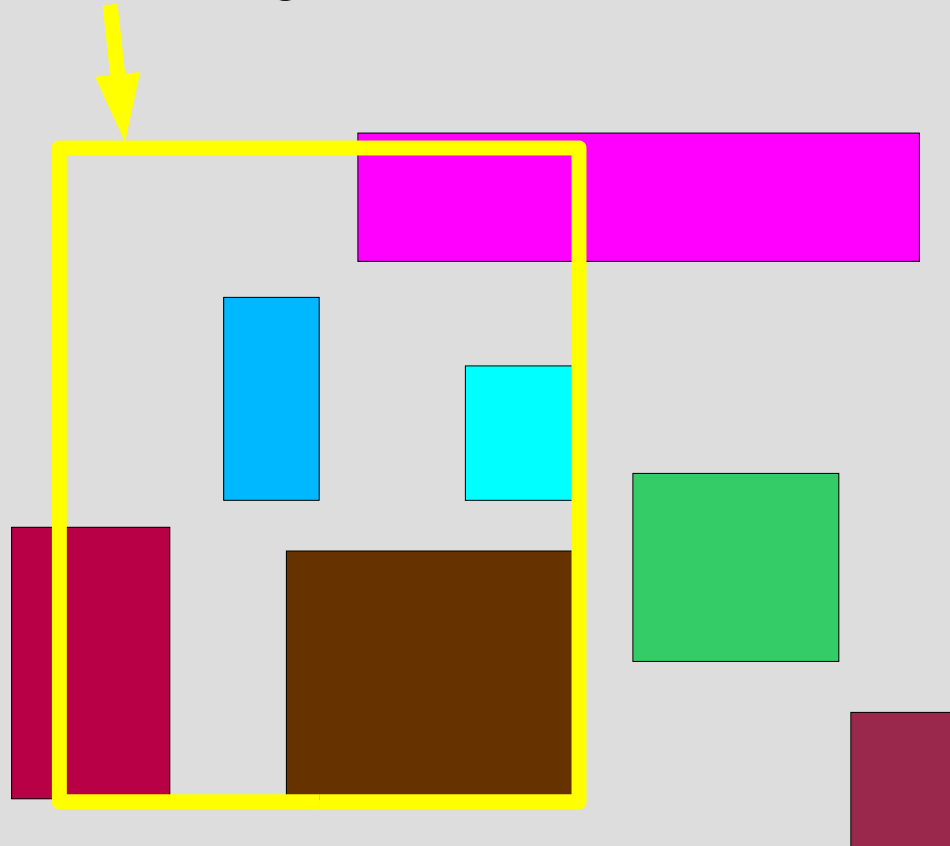
Es gibt Orte, an denen sich Objekte normalerweise nicht befinden,

z.B. bei Transportern: Seen (ohne Fähre usw.).

Ausschlußregionen

Idee:

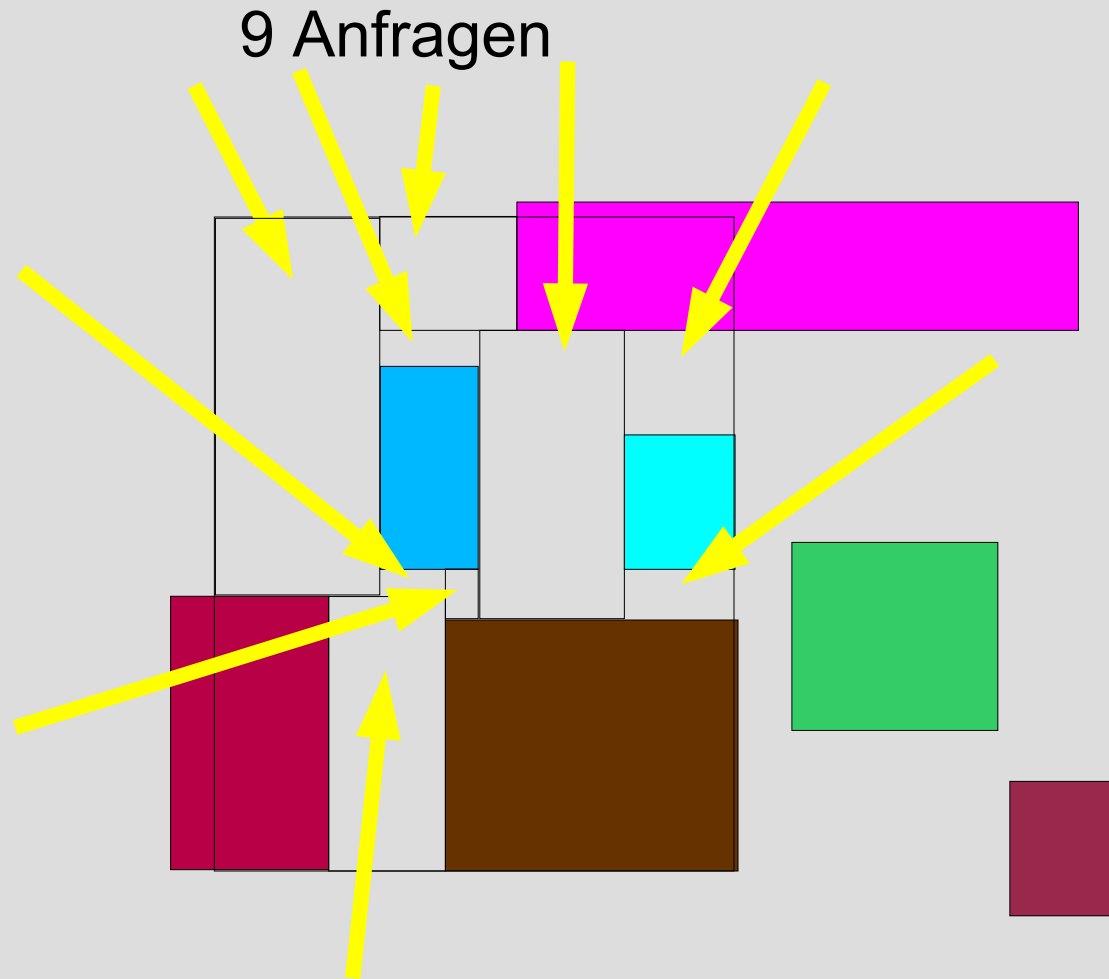
1 Anfrage



Anfragen müssen nicht den kompletten Bereich absuchen.

Ausschlußregionen

Idee:



100 Anfragen können in kleinere aufgeteilt werden.

Zusammenfassung

- B-Baum
 - keine effizienten räumlichen Suchanfragen möglich
- R-Baum
 - Erweiterung des B-Baum (mehrdimensionale Indexierung)
- R*-Baum
 - „Performance-optimierter“ R-Baum
- TPR-Baum
 - Anpassung des R*-Baum für „sich stetig fortbewegende Objekte“

Ende

Danke für die Aufmerksamkeit!

Fragen ?

Anhang

R-Baum ChooseLeaf

- 1) Steige im Baum bis zu einem Blatt ab.
- 2) **Wähle den Nachfolger** zu dem Verzeichnisrechteck, welches die **kleinste Vergrößerung** benötigt, **um den neuen Eintrag E zu überdecken.**
- 3) Wenn das nicht eindeutig ist, wähle das Rechteck mit der kleinsten Fläche.

Reinsert

Insert:

```
ChooseLeaf;  
  if (Überlauf)  
    Reinsert;  
  if (Überlauf)  
    NodeSplit;
```

Für alle Einträge eines Knotens:

Berechne die **Entfernung** zwischen den **Zentren der Rechtecke** und dem **Zentrum des MBRs** des Knotens.

Lösche die Einträge mit der **größten Entfernung**.

Füge die **gelöschten Einträge** beginnend mit dem entferntesten wieder ein (**Insert**).

NodeSplit

Insert:

```
ChooseLeaf;  
  if (Überlauf)  
    Reinsert;  
  if (Überlauf)  
    NodeSplit;
```

1)ChooseSplitAxis: bestimme Achse, an welcher der Split ausgeführt wird

2)ChooseSplitIndex: teile Einträge in zwei Gruppen entlang der Achse ein

3)Verteile der Einträge auf zwei Gruppen

R*-Baum ChooseSplitAxis

NodeSplit:

ChooseAxis;

ChooseSplitIndex;

Teile auf;

Für jede Achse:

Sortiere die Einträge zunächst nach dem **unteren**, dann nach dem **oberen Wert ihrer**

Rechtecke und bestimme alle Verteilungen. Berechne die

Summe aller Umfangswerte pro Verteilungen.

Wähle die Achse mit dem **minimalen Wert** als Splitachse

R*-Baum ChooseSplitIndex

NodeSplit:

ChooseAxis;
ChooseSplitIndex;
Teile auf;

Entlang der gewählten Achse,
wähle die Verteilung mit dem
minimalen Überlappungswert.
Falls es mehrere Möglichkeiten
gibt, wähle die Verteilung mit
minimalem Flächenwert.

TPR-Baum

- Erweiterung des R^* -Baums
- Algorithmen vom R^* -Baum werden übernommen, wobei die Funktionen über die Eigenschaften von BRs durch ihre Integrale ersetzt werden

$$\int_{t'}^{t' + H} A(x) dx, \text{ mit z.B. } A(x) \text{ ist die Fläche}$$

t' Zeitpunkt der letzten Änderung

H Horizont: Haltbarkeit des Index + max. mögliche Anfragezeit