Labo Algoritmen voor Beslissingsondersteuning

Gedetailleerde beschrijving vierde opdrachtlabo (en finale code)

In dit laatste labo geven we meer vrijheid aan de kranen. Waar in het vorige labo de kranen gebruik maakten van een reservatie-mechanisme en een deel van de yard enkel konden betreden wanneer de andere kraan daar niet was, zullen de kranen in dit labo volledig vrij kunnen bewegen, maar moeten ze een minimale veiligheidsafstand onderhouden. Gelijkaardig aan het vorige labo, zal dit in de praktijk leiden tot een potentiële efficiëntie-verbetering, opnieuw ten koste van een complexer algoritmisch model.

Waarop te letten:

1. Kranen staan op één enkel rail-systeem en kunnen elkaar dus niet voorbijsteken
2. Kranen mogen nooit botsen. Om botsingen te voorkomen wordt gebruik gemaakt van een veiligheidsafstand δ_x^{safe}, die als parameter meegegeven wordt - per kraan - in de instantie-file. Dat wil zeggen dat $x_{q_2} - x_{q_1} > \delta_x^{safe}$
3. Gezien er twee kranen zijn, dien je bij elke beweging die je in de output-file meegeeft ook het kraan-id q_i mee te geven.

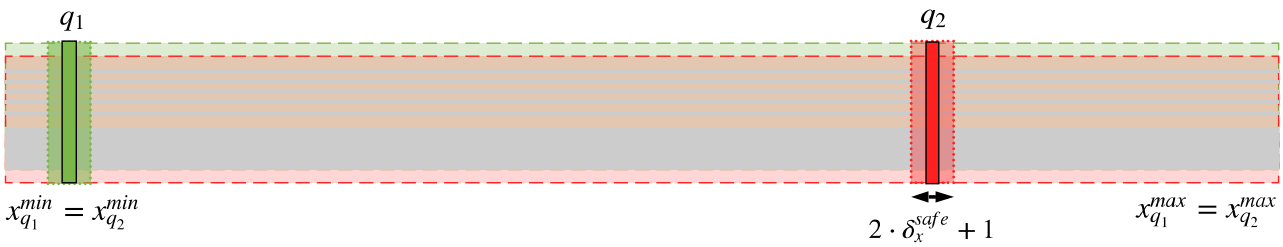

Figure 1: Twee kranen die vrij kunnen bewegen, maar een veiligheidsafstand moeten onderhouden

Fysische restricties

De fysische restricties zijn eigen aan de yard en containers en kunnen bijgevolg op geen enkel moment overtreden worden.

1. Je kan nooit hoger dan H_{max} stapelen
2. Je mag altijd langere containers op kortere zetten, maar nooit omgekeerd
3. Als je een langere op kortere stapelt, moet de lange container steeds volledig ondersteund zijn (bvb. twee types 20 ter ondersteuning van een type 60 is niet toegelaten).
4. De som van de lengtes van de ondersteunende containers moet even groot zijn als de lengte van de container die erop staat.

Veiligheidsvoorschriften (set R)

De veiligheidsvoorschriften zijn situatie-afhankelijk en gelden enkel op de uiteindelijke stapeling die je bekomt (dat is diegene die je naar de .yso-file wegprint). Tijdens het omvormen hoef je daar dus niet aan te voldoen.

1. $\forall stack \in s'' : issorted(stack) = true$
2. $\forall stack \in s'' : weight(top(stack)) > 1$
3. $\forall stack \in s'' : height(stack) \leq H_{safe}$

Nodige gegevens:

1. De reistijd van kranen is onveranderd, namelijk $v_x = 1$ en $v_y = 1$
2. Het opnemen of neerplaatsen van een container kost tijd. Reken hiervoor 2 tijdseenheden
3. Wanneer containers verplaatst worden is belangrijk. Geef dus in de output-file (.yso) bij de kraanbewegingen ook een container-id c_{id} mee wanneer je het opnemen of neerplaatsen van een container begint
4. Ook welke kraan een beweging uitvoert is belangrijk. Nu geef je dus de verplaatsingen van verschillende kranen tegelijk door. De verplaatsingen zijn chronologisch gesorteerd in de output-file en per lijn geef je nu als tweede entry het kraan-id q_{id} mee

```
# kraanbewegingen (t,q_id,x,y[,c_id])
0,1,0,0
0,2,40,0
7,1,0,7
20,2,20,6,7 # opnemen container 7
22,2,20,6
34,2,20,18,7 # neerplaatsen container 7
36,2,20,18
56,2,0,0
```

Te uploaden in één zip-archief per groep

1. De source-code van de door jullie ontwikkelde klassen
2. Een executable jar die voor een gegeven input-file een output-file genereert adhv het commando: `java -jar LabAB05.jar inputfilename.ysi outputfilename.yso`
3. De .yso-files voor de .ysi bestanden gegeven op Toledo.